

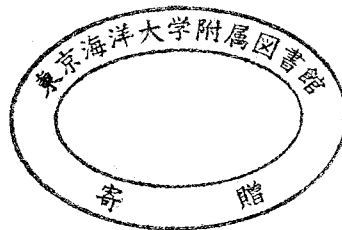
携帯電話の特徴を利用した船員のパーソナルコミュニケーションと人員把握に関する研究 : GPS内蔵型携帯電話と無線LANVoIPの利用

著者	梅野 智恵
学位授与機関	東京海洋大学
学位授与年度	2006
URL	http://id.nii.ac.jp/1342/00000834/

修士学位論文

携帯電話の特徴を利用した船員のパーソナルコミュニケーションと人員把握に関する研究
・GPS 内蔵型携帯電話と無線 LANVoIP の利用・

平成 18 年度
(2007 年 3 月)



東京海洋大学大学院
海洋科学技術研究科

海運ロジスティクス専攻
梅野 智恵

abstract

Many ships and their cargos are being managed safely by positioning report systems. However, little attention has been paid to the safety of the crew. Before, the present authors attempted an inboard system that used PHS. However, the functions were just voice call and mail exchange. Data acquisition from the terminal for proper control was not possible. Thus the position of the terminal was not available. As for next generation cell phones, GPS receivers and wireless LAN are installed by manufacturers. Therefore, a system is proposed which uses a cell-phone with a GPS receiver on a ship in order to promote the safety of ship's crew. The availability of a cell-phone GPS receiver was checked at thirty different points inboard. Positioning was not possible in the areas further than 4m from the window. Subsequently, a system which follows the positions of the crews and confirms their safety inboard by using VoIP (Voice over Internet Protocol) function by wireless LAN was proposed.

The VoIP (Voice over Internet Protocol) functionality by GPS and wireless LAN is installed in next generation cell phones. In this research, crew's safety on board is monitored by using the terminal of IP telephone with a GPS equipped cell phone and voice call to acquire the location information of each member.

Presently, ship navigation is conducted by the information that exists on the bridge. GPS receivers are usually used as a powerful tool to find positions. However, positioning information is just used for ship navigation and not for the safety of crew individuals. Thusly, the usage of a handy-type GPS receiver built in cell phone on a ship for monitoring the safety of the ship's crew is proposed. It is portable and works even inside rooms. Also investigated is the possibility usage area of a Cell Phone GPS. Personal communications and positional detection between crew members were done with a wireless LAN terminal that was available for use anywhere inboard.

In this paper, the performance of a cell phone which equip GPS receiver (CASIO A5403CA) and a Wireless VoIP terminal (ICOM VP-43) is examined and their possibilities to be a tool to secure the crew's safety is discussed.

学位論文要旨

海洋科学技術研究科 海運ロジスティクス専攻

0555006 梅野 智恵

これまでコードレス電話や PHS を船内で使う試みはされていたが、音声通話のみで端末個々の制御や情報の入手が自由にできず船員間の自由な情報交換や位置把握には利用できなかった。次世代の携帯電話には GPS 受信機や無線 LAN による VoIP(Voice over Internet Protocol)機能(IP 電話)を搭載している。そこで本研究では、位置情報の取得に GPS 携帯電話と音声通話に IP(Internet Protocol)電話の端末を利用して、船内における船員の安全確認を目的とする仕組みを提案する。

GPS 受信機は室内で位置を取得することに限界がある。そこで、GPS で測位のできない船内では無線 LAN 搭載の VoIP 端末とその IP アドレスを用いて船員間のパーソナルコミュニケーションと位置検出を提案する。本システムは IP 電話機器と連動動作しサービスが提供できる Asterisk(内線交換機)機能を使う。本研究では、まず GPS 内蔵型携帯電話(CASIO A5403CA)の船内における受信状況と測位精度の調査を行い、Asterisk の持つ SIP(Session Initiation Protocol)サーバ機能により、汐路丸船内を移動している時の無線 VoIP 端末(ICOM 製 VP-43)から Asterisk の SIP サーバへの登録手続きを一定間隔(最小1分間隔)で行い、VP-43 端末からの通信の可否の状態を記録する基礎実験を行った。

GPS 携帯電話を使用し船内 30 箇所位置情報を取得した結果、通常単独測位モードでは、2drms は 67.8m という結果になった。また、窓から 4m 以上離れると測位不可能であった。次に各 VP-43 端末(@192.168.10, @192.168.11)とアクセスポイント(@192.168.1.102)に IP アドレスをつけ、Asterisk へ SIP サーバの登録手続きの状態を 30 秒毎に出力した。このことにより、どこのアクセスポイントを使用して登録手続きが行われているかがわかる。つまり、その端末(VP-43)を持った船員がどのアクセスポイント付近にいるかが判断できる。船内にアクセスポイントを設置し、VP-43 端末の ON,OFF を自動的に知らせることで船員の安全が確認できる。

目次

1 序論	1
1.1 船員の位置把握装置の現状	1
1.2 研究の目的	2
1.3 GPS とは	3
1.4 GPS の概要	3
1.5 GPS 内蔵型携帯電話の概要	5
1.5.1 救急通報システム (Emergency 911) の搭載	5
1.5.2 携帯電話会社の対応	5
1.5.3 携帯電話 GPS アンテナの特徴	6
1.6 本研究に使用した GPS 内蔵型携帯電話	8
1.6.1 KDDI の A-GPS サービス	8
1.6.2 gpsOne の方式	8
1.6.3 GPS 内蔵型携帯電話の測位方式	10
1.6.4 GPS 内蔵型携帯電話のデータ取得の仕方	11
1.6.5 KDDI の携帯電話の利用可能な範囲	14
1.6.6 GPS 内蔵型携帯電話で取得した位置情報の活用	15
1.7 GPS 内蔵型携帯電話を用いた測位精度の実験	17
1.7.1 陸上における固定点の実験	17
1.7.2 陸上における移動時の実験	18
1.7.3 携帯電話の基地局と誤差の関係	22
1.7.4 船橋における測位精度	23
1.7.5 船内各所における測位精度	24
1.7.6 勝どき停泊中における測位精度	27
1.7.7 陸上からの安全管理について	28
1.8 まとめ	29
2 無線 LANVoIP	30
2.1 無線 LAN の概要	30
2.2 VoIP(Voice over Internet Protocol)	33
2.2.1 VoIP の概要	33
2.2.2 IP 電話システム内で活用する携帯型無線 IP 電話	37
2.2.3 NetStumbler を用いたネットワークの測定	37
2.2.4 Asterisk とは	41
2.2.5 SIP サーバを使用した実験	42
2.2.6 まとめ	44

3	結論	45
	参考文献	46
	謝辞	47
	付録	

1 章

序論

船舶が安全な航海を行うために日本では、船位通報制度 JASREP(Japanese Ship Reporting)があり、船舶の位置に関する情報を航行中の船舶から通報を受けて、コンピュータで管理し、24 時間毎の通報を遅延した船舶については、海上保安庁から船舶への呼び出し、及び海岸局、船舶所有者、代理店、付近航行船舶への問い合わせ等により安否を確認することを行っている。万一海難が発生した場合には、通報された情報から巡視船艇・航空機及び付近航行船舶の協力を得ることにより、迅速かつ効果的な搜索救助活動を可能とし、参加船舶の安全な航行にとって非常に有利な制度と言われている。JASREP の適用海域、参加船舶は、北緯 17 度以北、東経 165 度以西で、この海域にある全ての船舶（外航船からヨットまで）から通報を受けることになっている。近年においては、高まる海賊対策の一環として、また海上テロ防止の一環としてこのエリア外にある船舶からも通報を受けるなどしている。船舶全体に対して行っている安全対策はあるが、船員 1 人 1 人が位置情報を入手することが困難であることや、通信・通話料が高額であるため、各船員 1 人を対象とした安全確認や人員把握はできないでいる。

1.1 船員の位置把握装置の現状

船内での船員の位置把握装置は微弱無線、特定小電力無線機などの特殊小型発信機を利用したものや、PHS(Personal Handy phone System)、PHS 無線 LAN の PHS 無線機を利用したものが開発されている。しかし、試作段階で製品化されていないことや、小型化が困難であること、長時間の利用が難しいこと、送信のための許可が必要なこと、通話距離が思うほど得られないこと、情報を発信する間隔が自由に決められないこと、送信する情報の自由度がないことが問題点として挙げられる。

また、洋上の船内でのパーソナルコミュニケーション手段として、図 1.1 に示すように、船上通信装置や特定小電力トランシーバを利用している。船上通信装置は高価で、音声通信装置のみに使用されている。また、特定小電力トランシーバは、国内のみで使用が可能で利用できる範囲が狭いことが欠点として挙げられる。船内でのコミュニケーション手段は、コードレス電話や PHS を使用して行われている。最近では、PC(Personal Computing)によるネットワークを利用したコミュニケーションも行われるようになってきた。

そこで、現在、普及している携帯電話は人間のコミュニケーション手段として利用でき

ないかと考えた。次世代携帯電話の特色として、携帯電話機能の他に GPS 受信機や無線 LAN を使った IP 電話機能(Wireless VoIP (Voice over Internet Protocol):WVoIP)が内蔵される。既に一部で市販が開始され、事業所で利用されている。そこで、この機能を利用し、GPS 受信機により、船内や船外であれば、絶対的な位置と、無線 LAN により船内であれば、大まかな位置は把握が可能になるのではないかと考えた。図 1.2 が次世代の携帯電話の姿である。

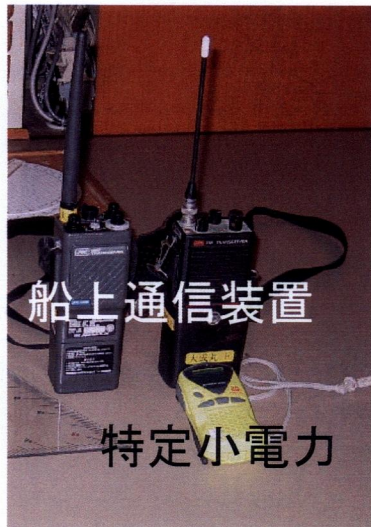


図 1.1 コミュニケーション手段



図 1.2 次世代携帯電話の姿

1.2 研究の目的

本研究の目的は次世代携帯電話の機能の検証を行うこと。GPS 内蔵型携帯電話において、船内や船外で位置の測位精度を測定すること、また、Wireless Voice Over IP(WVoIP 端末)を船員間のコミュニケーション手段として利用できるか確認すること。そして、船内で大まかな位置検出の仕組みとして、VoIP サーバ（ソフトウェア内線交換機）を使用して、サーバと WVoIP 子機通話間隔を確認すること。

そして、将来的な船員用の船内通信装置兼、電話機から、船外通信装置と WVoIP 端末、データ通信による船上での陸船間パーソナル通信の実現を目的として評価することである。

1.3 GPS とは

現在、位置情報を取得する手段として最も一般的に利用されているのは GPS である。GPS は Global Positioning System(全地球測位システム)の略で 1970 年代に米国国防省(DOD: Department of Defence)により開発が始められた人工衛星による測位システムである。1993 年末に正式的にシステムとして完成宣言が行われ、現在では 24 衛星の運用で全地球をカバーし、地球上どこでもいつでも高精度の 3 次元測位を可能とする測位システムとして、世界的に広く受け入れられている。もともとの主目的は米国の他、北大西洋条約機構(NATO)、カナダ、オーストラリアなどの同盟国の軍事用であるため、一部の民間用として公開されていたものには、測位の精度を故意的に劣化させる操作 (SA: Selective Availability) をしていたが、2000 年 5 月 2 日にその SA は解除され、それ以前は 100m 程度であった単独測位の精度は 10m 程度まで大幅に改善された。

また、モバイル通信の普及、GPS 受信機の小型化高性能とともに利用分野も大きく広がり、車、船舶、航空機等の移動局の位置を測位するだけでなく、地殻変動や地形の測量など、今後もその利用はますます増加すると考えられている。

現在では、GPS なしに陸、海、空に渡る交通輸送機関は安全で効率的な運航は危険であると言っても過言ではない程になっている。さらには地震予知などのための地殻変動監視システムは GPS を全面的に頼っている。

2005 年 12 月 15 日、ブッシュ大統領が新 PNT (Positioning:測位、Navigation:航法、Timing:タイミング) 政策の承認を発表し、この新しい PNT 政策により、GPS とその関連サービスの将来的な利用、管理および改良に関する方針が明確になった。

これは、ユーザへの GPS の可能性を確実にし、GPS 業界の明るい未来を約束するもので、この政策が GPS をグローバルな情報技術標準として安定させるための道を開き続けるということである。

10 年間で 2 度目の大統領声明として、PNT 政策は世界中の GPS ユーザからのダイナミックなニーズに応じるための安定したグローバルな政策環境の形成に関する米国の方針を示している。

1.4 GPS の概要

GPS システムは以下で構成されている。

- 1) 宇宙部分 (6 つの異なる軌道に 24 の GPS 衛星)
- 2) 制御部分 (5 つのモニター局、1 つの主制御局、3 つのアップロード局)
- 3) ユーザ部分 (GPS 受信機)

GPS 衛星は、衛星軌道の高度約 20,000km、傾斜角 55° 軌道間隔 60° の 6 軌道を回る周回衛星であり、各軌道に 4 個ずつの計 24 個で運用されている。現在では、GPS 衛星は 28 個の衛星が運用しており、周回の周期は 12 時間弱（11 時 58 分 02 秒）で、1 年で同じ配置に戻るようには推移している。実際には、軌道が随時変更されて、決まっていないが、衛星の軌道は 2km の範囲で管理されており、2 時間間隔で更新される高い精度を持った軌道情報を対応した衛星が 30 秒周期で放送している。

また、GPS では GPS タイムというものがあり、この時刻のもとに運用されている。各衛星には、セシウム原子時計または、ルビジウム時計が搭載されており、この原子時計は地上の官制システムで管理されており、この補正情報を軌道情報とともに衛星で放送している。GPS タイムは一週間で一回りするようになっている。

また世界標準時である UTC が知られているが、UTC と GPS タイムの違いも管理されている。表 1-1 に GPS 衛星の主要諸元を記す。

表 1-1 GPS 衛星の主要諸元

衛星個数	4 個 × 6 軌道面 (4 個補助)
軌道半径	26.561km
衛星設計寿命	7.5 年
周回周期	12 恒星時間[約 11 時間 58 分 02 秒]
軌道傾斜角	55°
搬送波 周波数	L1=1,575.42MHz(10.23MHz × 154) L2=1,227.6MHz(10.23MHz × 120)
測距信号	C/A code:L1 で送信、民間にも開放 P(Y) code:L1,L2 で送信、非公開
送信電力	L1:C/A Code:約 26W P Code:約 13W L2:P Code :約 4W
測位精度	水平方向約 10m
時刻精度	C/A code:約 20ms

各衛星は 2 つの L 帯周波数（L1 帯）は、C/A コード（Coarse Acquisition Code）と P コード（Precise Code）、L2 帯は P コードを 2 相位相変調して搬送する。C/A コードはコードパターンが公表されているために利用できるが、P コードはその周期が間引きなどで、一週間に合わされていて、民間利用者向きではない。

現在では、軍事用の P コードが公表されたため、更に秘密性の高い Y コード(P コードに含まれる情報を暗号化したもの)が開発され米国関係者のみが利用可能となっている。全衛星が同じ周波数で送信しているに関わらず、混信しないのは、両波ともにゴール

ド符号と呼ばれる擬似ランダム雑音（PRN:Pseudo Random Noise）で BPSK 変調（Bi-Phase Shift Key:2 相位相変調）されており、衛星ごとに異なる符号パターンを割り当てることで、どの衛星かを識別できるからである。

1.5 GPS 内蔵型携帯電話の概要

1.5.1 救急通報システム(Emergency 911)に向けて

2007 年の 4 月から、ついに日本版 E911 の携帯電話からの緊急通報時における位置情報の通知化が義務付けられることになる。総務省では今年 1 月に改正した事業用電気通信設備規則において、「緊急通報に関する機能」を追加し、事業用電気通信設備が通報者の場所を管轄する警察機関等への接続と、発信者の電話番号および位置情報の通知機能を備えるように規定している。すでに米国では、法制化により、北米携帯電話による救急通報システム(Emergency 911, 略称 E911)の搭載が義務付けられ、携帯電話事業者各社において各種の測位が検討されている。この結果、米国クアルコム(Qualcomm)社で開発された CDMA(Code Division Multiple Access,符号分割多元接続)方式の携帯電話では、測位精度が優れ CDMA 方式に適する GPS が、E911 の測位システムとして採用された。

施行期日は平成 19 年 4 月 1 日とされており、これらにより携帯電話・PHS からの緊急通報時に通報者の電話番号に加え、その位置情報が自動的に警察や消防などへ伝えられることになる。特に第 3 世代 (3G) 携帯電話については、原則的に GPS による位置情報を使うこととされており、電話番号に加え緯度・経度・測定誤差情報が通知されることになる。施行当初は東京、神奈川、愛知、大阪などで運用を始め、翌年春には全国 52 箇所にある 110 番通報の受信拠点のうち、半数へ導入されることが目標とされている。

1.5.2 携帯電話会社の対応

平成 18 年 12 月末の携帯電話の契約数は全事業者合計で 9490 万契約を突破(社団法人電気通信事業者協会データより)しており、今後これらの携帯電話すべてに GPS 機能が搭載されていくことが予測される。

既に KDDI(au)では販売している端末の 9 割以上が GPS に対応しており、NTT ドコモやソフトバンクモバイルでも、新モデルから GPS 対応モデルが増加している。

衛星測位技術は、「衛星からの信号を受信して測位する」というその性質のため、衛星との視野が確保できる屋外環境での使用が前提となり、建物や障害物により信号が遮断される屋内環境には不向きである。近年、新たな信号処理技術や AGPS (Assisted GPS : 外部支援型 GPS) 方式などの採用によって、受信機の感度が向上し室内環境でも GPS 信

号を受信し、測位できる受信機も出てきているが、これらいわゆる高感度制の GPS 受信機の場合、衛星からの直接波だけでなく複雑に反射したマルチパスも受信するため、たとえ測位が可能であったとしても、精度が悪化するケースがある。また、微弱な信号を積算処理する場合、屋外と比べて測位に時間がかかるといった問題もある。さらに、地下街や窓がない建物の内部など、高感度型の GPS 受信機を使っても依然として測位不能となる環境が多いのが現状である。

1.5.3 携帯電話 GPS アンテナの特徴

携帯電話に内蔵される超小型 GPS 用チップアンテナが開発された。開発品は、高誘電体セラミックスを母材として小型化を図り、逆 L 型放射電極にインピーダンス整合のための給電線を一体化した独自の構造を有する。携帯電話に GPS を搭載し、歩行ナビゲーション機能として利用するほか、緊急通報支援システムなどと連動し、個人の位置情報をリアルタイムに通知するサービスの利用にも用いられている。表 1-2 に GPS アンテナの代表的な仕様例を示す。

一般的に商業利用で受信される GPS 信号の搬送波の中心周波数は 1575.42MHz、帯域幅は 2.046MHz である。アンテナから GPS 信号を安定に受信するため、高い利得が望まれるが、携帯電話用アンテナとしては、端末の姿勢によらず、複数の衛星からの信号を受信する必要があることから端末周囲にできるだけ均等な利得分布（無指向性）を持つことが特に要求されている。

表 1-2 GPS アンテナの目標仕様例

項目	仕様
周波数帯域[MHz]	1,575.42±1.023
平均利得[dBic]	-5 以上
偏波方向	右旋円偏波
指向性	無指向性
電圧定在波比(VSWR)	2 以下
入力インピーダンス[Ω]	50(Typ.)
外形寸法[mm]	6(L)×4(W)×4(H)

従来、車載用 GPS アンテナなどでは、パッチアンテナが主に使われている。GPS 衛星

から発する右旋円偏波の電界成分をパッチ(Patch)と呼ばれる放射電極で感知することにより、高い利得が得られるためである。しかしながら、パッチアンテナはチップ外形寸法（一辺 25mm 程度）が大きいので、そのままでは携帯電話のような小型端末への搭載には適さないほか、放射電極と垂直方向に利得が偏在するため、無指向性を満足にできない。アンテナ基体の材料である誘電体セラミックスの比誘電率を ϵ_r 、搬送波の波長を λ [m] とすると、誘電体の波長短縮効果により、放射電極の一辺の長さ L [m] は概略次式で表される。

$$L = \lambda / 2 \sqrt{\epsilon_e} \quad (1)$$

$$\epsilon_e = (1 + \epsilon_r) / 2 \quad \epsilon_e: \text{実効比誘電率} \quad (2)$$

上式からアンテナを小型化するためには、誘電体の比誘電率を大きくすればよいが、これに伴い帯域幅が減少し、さらに利得が低下することから、パッチアンテナの小型化には限界がある。

一方、携帯電話用 GPS アンテナは図 1.3 に示すような 1 本のホイップアンテナを共用する方式もある。この場合、携帯電話回線に用いられるセルラの送受信帯域(CDMA 方式では一般に 800MHz 帯使用)と GPS 帯域のアンテナを共用するため、アンテナの後段に共用器、あるいはスイッチを設けることにより、それぞれの周波数帯域に分離される。

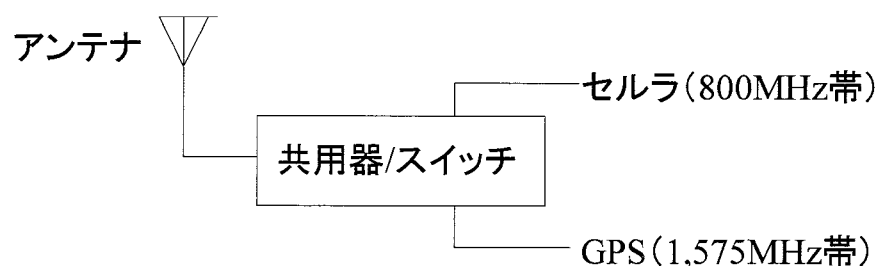


図 1.3 GPS アンテナ共用方式

表 1-3 に、アンテナ方式の比較例を示す。GPS 専用アンテナを用いることにより、共用器やスイッチを省いて回路構成を簡素化し、アンテナ直下での挿入損失を著しく低減できるため、GPS の受信感度の増加やセルラの通話性能向上を図りやすくなる。反面、高利得を維持しながらアンテナの小型化や占有面積の低減を図ることが、アンテナ設計技術における大きな課題であり、独自のアンテナ構造を立案することによって課題を解決した。誘導体セラミックスを用いたチップアンテナの最大の特徴である小型化、とりわけ占有面積の低減に着目し、ブルートゥース用アンテナや GPS 用アンテナ製品の開発が

行われた。

表 1-3 アンテナ方式の比較例

	GPS 専用方式	共用方式
帯域分離	アンテナ	共用器/スイッチ
回路の簡素化	○	△
挿入損失	○	△
利得	(○)	○
占有面積	△	○

1.6 本研究に使用した GPS 内蔵型携帯電話

1.6.1 KDDI の A-GPS サービス

本研究に使用した GPS 内蔵型携帯電話は、KDDI (au) の CASIO 製 A5403CA を使用した。日本では au が 2001 年 12 月からサービスを開始した。Eznavigation サービスを利用した位置情報にて提供されている。GPS 内蔵型携帯電話のような小型端末で GPS を利用するには制約がある。ネットワークを利用して端末の負担を軽減するネットワークを利用して端末の負担を軽減するネットワーク支援型 GPS (A-GPS) という測位方式を採用することにより、電力の消費を抑えている。さらに電波の悪い環境でも使用することが可能になった。GPS 受信機で測位する場合は、GPS 衛星からの情報を受けて、GPS 受信機で位置を計算する。ネットワーク支援型 GPS (A-GPS) では、GPS 衛星からの情報はいったんネットワークを経由してサーバへ送られ、サーバで位置を計算し、その情報が携帯電話に戻ってくる。au の GPS 内蔵型携帯電話は A-GPS 機能を持っていて、この機能は au 独自の機能である。今回は位置情報以外の測位データを利用するために au の GPS 内蔵型携帯電話を使用することで、GPS 衛星の電波が届かないところでも携帯電話の基地局を使用することによって、位置情報を得ることができるため、海上では有効であるのではないかと考えた。

1.6.2 gpsOne 方式

gpsOne は、「cdmaOne」の Qualcomm 社が開発した、携帯端末の位置を知るための技術で、現在でも PDC 方式(Personal Digital Cellular system:デジタル携帯電話)の携帯電話や PHS では、電波をやりとりしている基地局を利用して端末の位置を知る位置情報サービスなどを提供しているが、この gpsOne は cdmaOne および cdma2000/1X で使われる。仕組み的には、gpsOne はその名の通り、端末の位置を計測するために GPS と携帯電話のシステムの両方をハイブリッドに利用しているのが特長である。

携帯電話の従来の位置情報と gpsOne のそれとの大きな違いは精度と言われている。

gpsOne は GPS の情報を利用しているため、非常に高い精度で携帯電話の位置を知ること

ができる。Qualcomm 社が配布している資料によれば、携帯電話のみを使った位置計測ではだいたい 300 メートル前後の精度でしか位置がわからないため「街のどのエリアにいるか？」程度しかわからないが、GPS との携帯電話の位置情報システムの組み合わせでは 5 メートルから 50 メートル程度の精度でわかるようになるため「どの通りのどの角か」といった情報を伝えられるようになるので、緊急通報に利用するなどということも十分可能になると考えられている。

互いに補完しあう GPS と携帯電話のネットワーク gpsOne では位置情報を得るのに、GPS 衛星からと携帯電話のネットワークから、両方の情報を利用し、情報をそれぞれ補完することによってより確かな位置情報を得ている。

携帯電話は仕組み上、電波をやりとりしている基地局を使って、ある程度の精度で自分のいる位置を知ることができるので、gpsOne ではこれを使って人工衛星からの電波が届いていないときでも、自分の位置を知ることができるようになっている。また、GPS と cdmaOne システムの補完のメリットとしては次のことが挙げられる。

cdmaOne のメリット

GPS では、最初に起動する際に、端末が衛星を検索して捕捉しなければならないため多少時間がかかることがあるが、gpsOne では、この作業を cdmaOne ネットワーク上のサーバがこの作業を補佐してくれるためこの作業が非常に高速化される。条件によってはものの数秒で衛星を補足することができる。

cdmaOne というシステムは、実は元々端末と時刻情報の同期を取るために、基地局側では GPS からの電波を使用しており、cdmaOne と GPS との連携は全く初めてというわけではない。

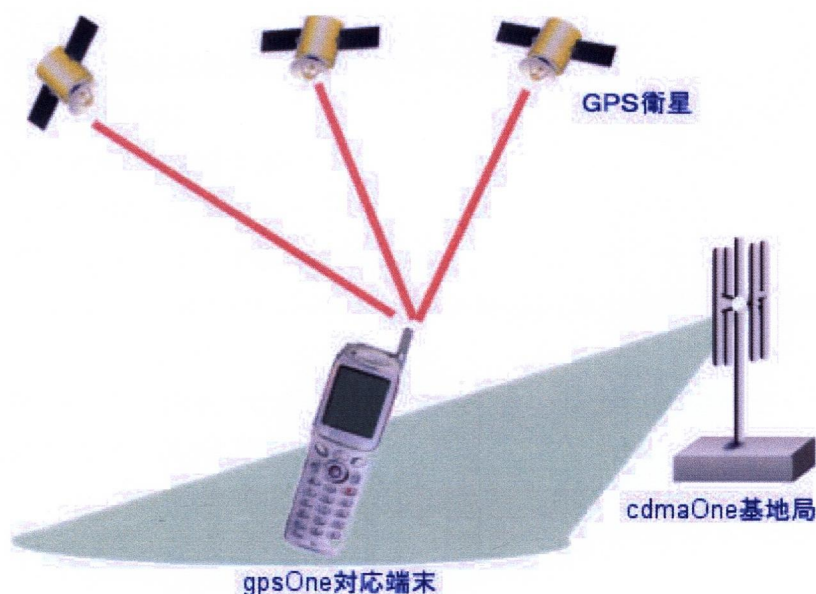


図 1.4 gpsOne 方式

gpsOne にはさまざまな応用が期待されている。中でも一番の注目は、アメリカの場合は「緊急通報」だとされている。日本でも問題になっていることで、従来はたとえば警察に 110 番通報する場合、システム上、従来は携帯電話では警察には電話のある位置がわからないために事故の起きた場所などがわからなくなってしまい、出動に支障をきたすことがある。

米国でも同様のことが起こっていたが、連邦通信委員会（FCC）の（Emergency:E-911）Phase II という規定ができ、緊急通報である 911 番では携帯電話からかかってきた電話でもその位置が誤差 50 メートル以内でわかるようなシステムが採用されることになった。そのため、携帯電話から位置情報が自動的に通知できるこの gpsOne が注目されている。

日本では、4 月 1 日よりセコムが開始している防犯サービス「ココセコム」がこの gpsOne を利用している。セコムの提供している専用端末に、この gpsOne をサポートしたチップが搭載されていて、KDDI(au)の cdmaOne のネットワークを使ってこの端末の位置がセンターに登録されている。これでインターネットを使って、特定の車両の位置を調べたり、セコムの緊急対処員を現場に派遣したりするサービスが行われている。

1.6.3 GPS 内蔵型携帯電話の測位方式

GPS 内蔵型携帯電話の測位方式は大きく分けて 3 つの方式がある。図 1.5 で示す。GPS 衛星のみを使用して位置情報を取得する方式と、GPS 衛星と基地局の情報を使用して位置情報を取得する方式の HYBRID 方式、携帯電話の基地局の情報を使って、位置情報を取得する AFLT 方式と PREFIX 方式と SECTOR-CENTER 方式がある。

- ・ AFLT_FIX : 基地局情報のみ
- ・ FAIL : 測位の失敗
- ・ FREE : 精度の指定をしない
- ・ GPS_FIX : GPS 情報のみ
- ・ HYBRID_FIX : GPS と基地局情報
- ・ PREFIX_AFLT : BTS(Base Transceiver Station)利用モバイル通信網（主に携帯電話）における基地局を使用した方式
- ・ SECTOR_CENTER : セクターセクター測位、基地局の指向性アンテナがカバーしている区域の中心点を使用した方式

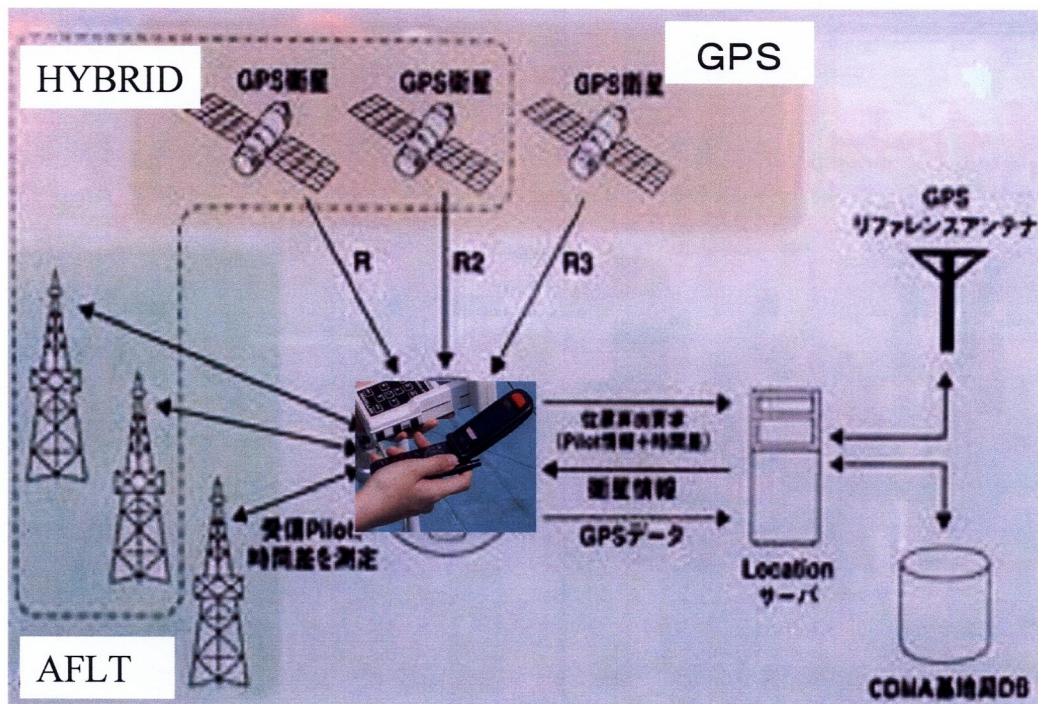


図 1.5 GPS 内蔵型携帯電話の測位方式

1.6.4 GPS 内蔵型携帯電話のデータ取得の仕方

通常、GPS 内蔵型携帯電話の機能で、GPS 情報を取得すると、図 1.7 に示すような地図の画面上にユーザの位置がわかるようになっている。これを数値的に評価する為に、図 1.8 に示し、公開されている GPS 情報をデータとして取得するため Java アプリを使用した。

1) Java アプリ開発

Java 言語はアメリカのサンマイクロシステムズ社が開発したプログラム言語で、C++ 言語をもとに開発された。そのため誰にでも簡単にプログラムが作成できるようになっている。そこで GPS 内蔵型携帯電話にて GPS 機能を使い、連続的に測位する Java プログラムを作成した。

2) 開発環境

まず、Java を作成する前に、PC の環境を整えなければならない。必要なものは

- ・ J2ME(java2 Micro Edition)
- ・ J2SE(java2 Standard Edition)
- ・ ezplus 開発キット

この 3 つの環境が必要である。これは Sun Microsystems 社のプログラミング言語で無料ダ

ウンロードできる。

3) プログラム手順

- ・ メモ帳でプログラムを作成し、Java ファイルとして保存する。
- ・ ezplus 開発キットの中にある、ezplus Tool をメモ帳で作成した Java 実行ファイルを作成する。その時、自動的に jar ファイル、class ファイル、kjsx ファイルが作成される。
- ・ ezplus 開発キットの中にあるエミュレータで作成された kjsx ファイルを実行確認する。
- ・ エミュレータ図 6 で実行できたならば、携帯電話に配信する。

3-1) MIDP

MIDP は携帯電話などの端末用に定義された Java の実行環境仕様プロファイルの一つで携帯電 J2ME 自体に加えてこのプロファイルが定義するライブラリを用意しなければならない。また MIDP 用に作られたアプリケーションの形式を MIDletj と呼び、Java アプリケーションのひとつで、ネットワークを通じてデータをダウンロードし、携帯電話上で動作させることができる。Java プログラムの一種であり、携帯電話の使用するための Java アプレットである。

3-2) KDDI ファイル

au が ezplus を用いる際には MIDP の基本 API の他に au 独自拡張の API が必要で。その拡張 API が KDDI のプロファイルである。KDDI プロファイルは各フェーズで異なり、それぞれで使用する異なる。現在ではフェーズ 1、フェーズ 2、フェーズ 2.5 がありフェーズの数値が増すたびに機能も増えている。

3-3) Java2 Micro Edition

J2ME はプログラミング言語「Java2 の機能セット」のひとつで電化製品や携帯情報端末、携帯電話などの組み込み機器むけの機能である。この J2ME の中に J2ME CLDC が組み込まれている。

3-4) J2ME CLDC

J2ME CLDC は携帯電話などの処理能力やメモリの制限のある機能を対象としているコンフィギュレーションのことである。サポートするクラスの数が必要最小限に絞り、クラスライブラリ自体のサイズを減らし、仮想マシンが実行に使用するメモリを小さくする機能を持っている。

3-5) KJX ファイル

実行ファイル (JAR ファイル) と属性ファイル (JAD ファイル) を結合する際に必要になるファイルである。サイズは最大で 50 バイト、データ保存領域 (レーダストア) のサイズが最大 10 k バイトとなっている。

4) プログラム作成手順

Jbuilder を用いて作成する際の手順は以下の通りである。

4-1) JDK の設定

JDK は Java 言語でプログラミングを行う際に必要最低限のソフトウェアのことである。

これは、Java の開発環境元である会社が無料で配布している。コンパイラ、デバッガ、クラスリブラリ、Java プログラム実行環境などが含まれている。

4-2) プロジェクト作成

プロジェクトとは、実行プログラムを作成するために必要な情報をまとめたファイルのことを言う。Java でいうとソースプログラムやインクルードファイルが存在するパスやオブジェクトの依存関係、必要なライブラリなどが記述されている。コンパイラをコマンドモードから実行したい時は、メイクファイルと呼ばれているテキストファイルを同様の目的で使用したが、プロジェクト名をつける。

4-3) ソースコード作成

プログラミング言語を用いて、記述したソフトウェアの設計図（ソースコード）は、そのままではコンピュータ上で実行することができないため、コンパイラなどのソフトウェアを用いてオブジェクトコードと呼ばれるコンピュータの理解できる形に変換して実行する。

4-4) クラスの作成

クラスを作成することにより、プロジェクトのプログラミングにおいてもデータとその操作手順であるメソッドをまとめたオブジェクトを定義することが可能となり、同時に同種のオブジェクトをまとめて取扱うことも可能になる。ここで定義することによりさまざまな機能を利用することができる。

4-5) ファイルの作成

JAD ファイル、マニフェストファイル、クラスファイル、JAR ファイル、bat ファイルを作成する。

5) エミュレータで実行

PC 上での JBuilder の中にあるエミュレータ図 1.6 で実行し、確認するこれは ezplus の開発キットエミュレータでは実行不可能である。

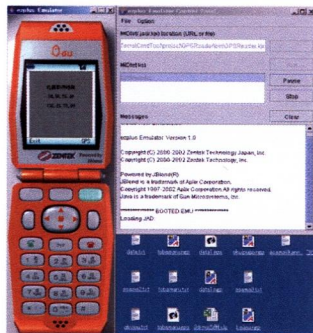


図 1.6 エミュレータ

6) 端末で実行する

パソコンのアップロードし、携帯電話に配信する。

このプログラムを作成し、携帯電話に配信することによって、通常は図 1.7 となるが、図 1.8 のように、電界強度、緯度、経度、バッテリーの残量、位置情報の取得に使用した基地局の緯度、経度、測位モードが表示できる。

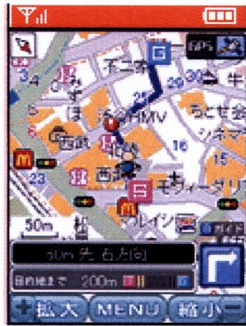


図 1.7 表示される画面

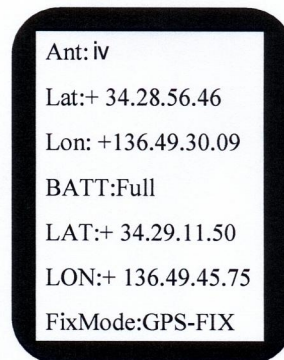


図 1.8 評価するために表示した画面

1.6.5 KDDI の携帯電話の利用可能な範囲

本研究に使用した、KDDI の携帯電話の利用範囲はインターネット上で、公開されている。http://www.au.kddi.com/service_area/index.html

今回、実験に使用したのは、関東エリアの東京湾内であるが携帯電話のエリア内である。

図 1.9 に利用可能な範囲を示す。日本の沿岸から青色で示したものが携帯電話エリア内を現している。およそ陸から 20～30 マイルまでは、通信が可能である。東京湾に関しては、通信可能な範囲になっている。



図 1.9 利用可能範囲

1.6.6 GPS 内蔵型携帯電話で取得した位置情報の活用

GPS 内蔵型携帯電話で取得した位置情報を活用する為に、連続的に測位することや取得したログを保存し、違うサーバから位置情報を見ることが GPS 内蔵型携帯電話を利用してできる。HDML 言語を用いて、`device:gpsone` にアクセスすることで、連続的に測位することや位置情報を閲覧することができる。

HDML 言語とは、Handheld Device Markup Language の略で、一般に表示機能に乏しい携帯端末向けのコンテンツ記述言語のことである。au と TUKA の EZWeb で用いられている。HTML と同じように、テキストの中のタグに埋め込んでコンテンツを記述するマークアップ言語である。1 枚のページカードという単位で扱い、複数のカードをまとめたものをデッキと呼ぶ。表示はカード単位で行い、データの送受信はデッキ単位で行われる。

Phone.com 社（旧 Unwired Planet 社）などによって提唱されたもので、同じく携帯端末向けの記述言語である Compact HTML とは互換性がない。

1.6.6.1 設置

- ・準備

cgi が使えるサーバを用意する。 cgi を設置するディレクトリでの GoogleMapsAPIKey を取得する。(参照→「GoogleMap を使う」)

- ・設置

cgi スクリプト中の「各サイトでの個別設定」の部分を設定する。 cgi スクリプトを設置する。(cgi スクリプト自体の文字コードは utf-8 を推奨) 設置するディレクトリ、このプログラムのパーミッションを適切に設定する。(ディレクトリはファイル書き込み可能、このプログラムは実行可能に。)

1.6.6.2 使い方

- ・ログを取る方法

au の携帯電話から MainMenu で、ログファイル名と通知間隔を設定
(ログファイル名と通知間隔をフォームに入力し、Execute のボタンを押す。)

au の携帯電話から StartLog のリンクに飛ぶ。

- ・ログを読む方法

MainMenu で、「ReadLog」のリンクに飛び、LogMenu から、読みたいログを選択する。
(GoogleMap の動く PC 上のみから可能)

1.6.6.3 CGI (Common Gateway Interface)とは

Web サーバが、Web ブラウザからの要求に応じて、プログラムを起動するための仕組みである。

従来、Web サーバは蓄積してある文書をただ送出するだけであったが、CGI を使うことによって、プログラムの処理結果に基づいて動的に文書を生成し、送出することができるようになった。CGI はどのような開発言語でも使用できるが、実際には Perl などがよく使われる傾向にある。類似の技術には SSI や ASP などがある。

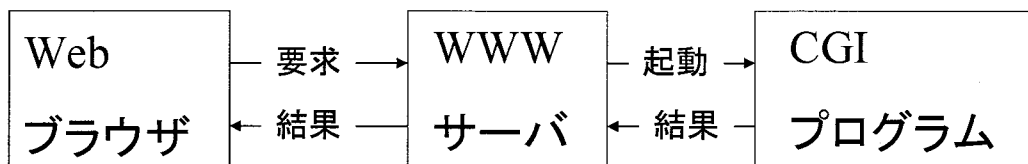


図 1.10 CGI の原理

1.6.6.4 位置情報の活用仕方

GPS 内蔵型携帯電話で取得した位置情報を他の相手にも伝えることができる。この動作

の流れとして、GPS 携帯電話で位置情報を取得→メール機能を使用して自分が伝えたい相手に送信→相手がメール受信→受信した相手が送られてきたメールから相手の居場所を判断するということができる。

また、今回は、図 1.11 に示すように陸上からの管理という視点から Web 上で、www サーバからユーザの位置情報を確認することを行った。

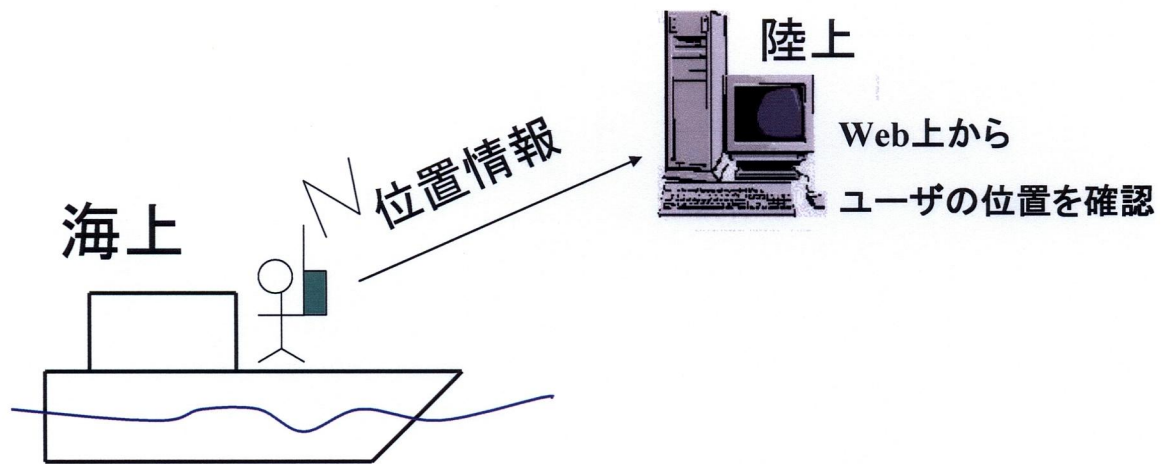


図 1.11 位置情報の活用方法

1.7 GPS 内蔵型携帯電話を用いた測位精度の実験

1.7.1 陸上における固定点の実験

まず、海上での実験の前に、陸上の固定点においてどの程度の精度が得られるのか実験を行った。実験した場所は、東京海洋大学第 4 実験棟の建物である。ここには、屋上に既知なアンテナがあることから、このアンテナを基準として測位精度を評価した。はじめに、屋上での固定点の測定を行い、建物 5 階の室内、窓から 0.5m のところにおける実験を行った。屋上においては、10 回測定し、室内については 30 回の測定を行い、測位精度を評価した。測位モードは全て GPS 測位モードである。

実験した結果を図 1.12 に示す。この結果より、室内における 2drms は 23.4m、屋上における 2drms は 22.6m となった。室内、室外ともに誤差においては、大きな変化はないと考えられる。また、今まで室内において、GPS 受信機を使用して位置情報を取得できなかったことを考えると、GPS 内蔵型携帯電話は、室内において位置情報を取得することが可能である。

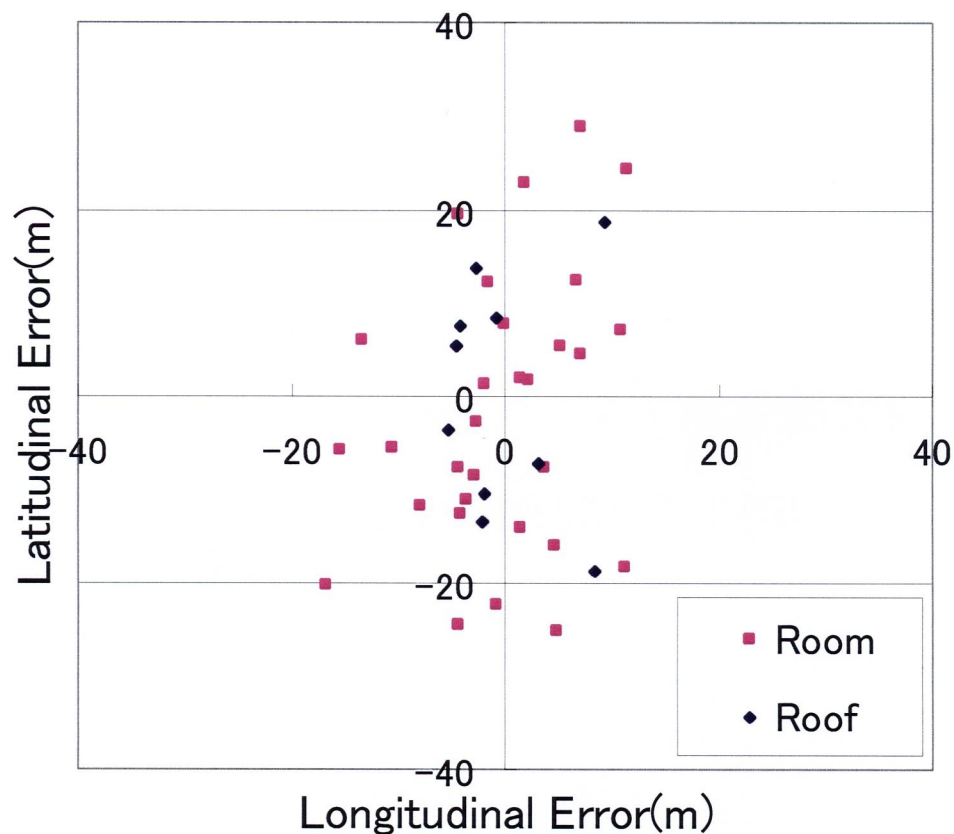


図 1.12 陸上の固定点における測位精度

1.7.2 陸上における移動時の実験

次に陸上において移動した場合の測位精度評価を行った。固定点に比べると、移動することによって、どの程度、測位精度が変化するか実験を行った。陸上における固定点については、実験を行ったが実際に使用しようと考えた、海上における船舶は移動体になっている。今回は、GPS 内蔵型携帯電話と、将来 GPS 内蔵型携帯電話に内蔵されると言われている、高感度 GPS 受信機(HGU7000-HD HITACHI 製)も使用した。今回、使用した高感度 GPS 受信機の特徴は、以下の通りである。

GPS 受信部

- ・受信周波数 L1 (1575.42MHz) C/A コード
- ・受信方式 スケーラブルコリレータ方式
- ・受信感度 -153dBm 以下
- ・測位更新レート 可変、最短 1 秒
- ・測地系 WGS84 測地系、および日本測地系

電池寿命

- ・測位／通信時 約 8 時間
- ・待機時 約 240 時間

実験は、実際に船の速度と同じぐらいの時速 30km で走行した。実験の様子は図 1.13 に示す。基準になるように高精度 GPS 受信機の NovAtel 社製の OEM4 受信機(図 1.14)を使用した。実験で走行した場所は図 1.15 に示す。東京海洋大学から相生橋を通り、勝どきのビル街を通り、晴海を通り、月島を通り、東京海洋大学に戻ってくる道のりで実験を行った。



図 1.13 実験の様子



図 1.14 基準にした OEM4 受信機



図 1.15 実験で走行した図

OEM4 を基準として、GPS 内蔵型携帯電話と高感度 GPS 受信機の測位精度評価を行った結果の航跡図を図 1.16 に示す。高精度 OEM4 受信機と高感度 GPS 受信機は実験の途中、受信できない場所もいくつかあったが GPS 内蔵型携帯電話は、常に位置情報を取得することが可能であった。高精度 OEM4 受信機を基準としての誤差を見た場合、図 1.17 の結果が得られた。高感度 GPS 受信機では、2drms は 9.1m となり GPS 内蔵型携帯電話の 2drms は 66.9m という結果になった。GPS 内蔵型携帯電話の測位モードは全て、GPS 測位モードの結果である。固定点の測位精度に比べ、精度が悪くなる結果となった。また、高感度 GPS 受信機は、受信する環境が大きく関わることがわかった。

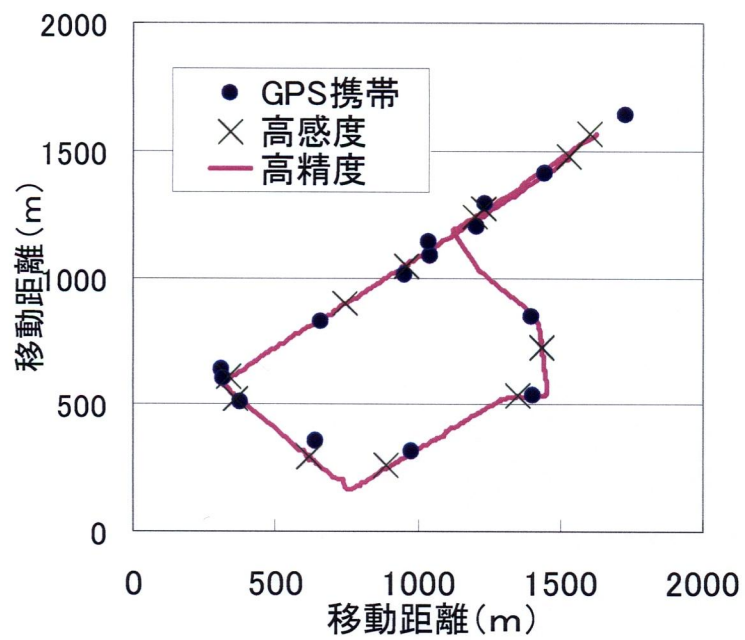


図 1.16 航跡図

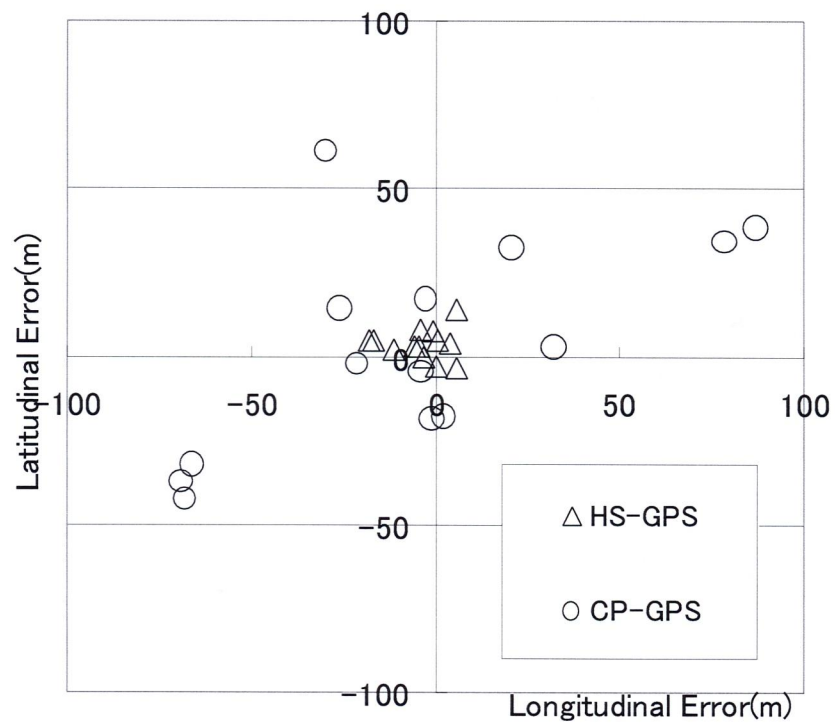


図 1.17 陸上の移動体における測位精度

1.7.3 携帯電話の基地局と誤差の関係

GPS 内蔵型携帯電話の特徴として挙げられる、基地局を使った測位モードがある。そこで、基地局を使って測位した場合の測位精度はどのように変化するのかを実験した。実験は、東京海洋大学の練習船汐路丸（図 1.18）を使用して実験を行った。



図 1.18 実験に使用した練習船汐路丸（全長 49m 425ton）

実験は、勝どき橋～館山湾までの航海の際に、位置情報を取得する際に使用した携帯電話の基地局までの距離と誤差の関係を調査した。汐路丸に設置されている DGPS(NAVCOM SF-2050G)を基準に、測位モード HYBRID モードのみを使用し精度評価を行ったところ、図 1.19 のようになった。位置情報を取得する際に使用した携帯電話の基地局までの距離と誤差の関係はないことがわかった。

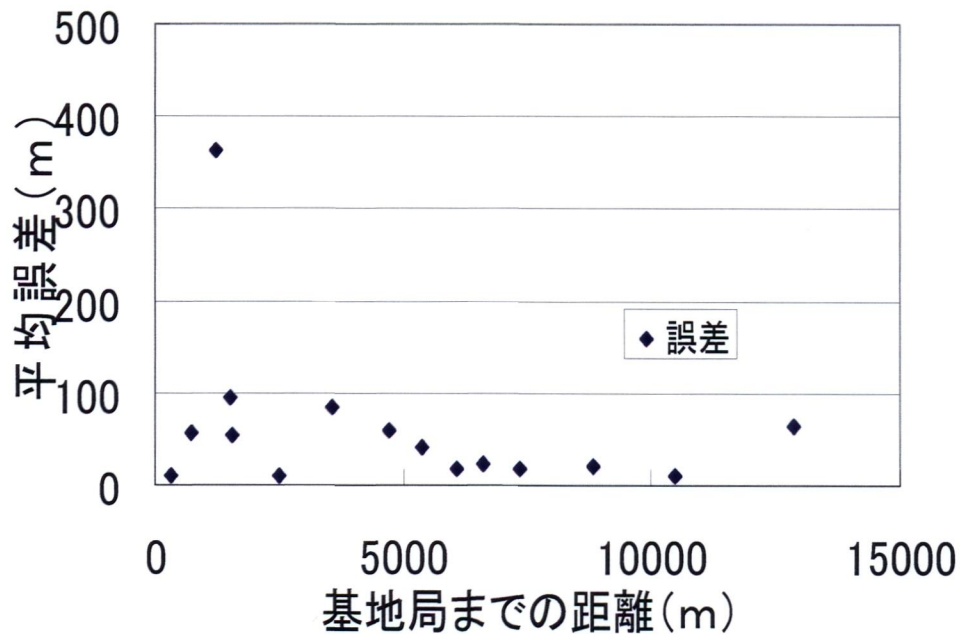


図 1.19 基地局までの距離と誤差の関係

1.7.4 船橋における測位精度

東京海洋大学練習船、汐路丸での航海中（館山湾）に、船橋において GPS 内蔵型携帯電話の測位精度の評価を行った。汐路丸に設置されている DGPS 受信機(JRC JLR-7700)を基準に行った。

その結果を図 1.20 に示す。船橋における測位精度では、**2drms** が **57.9m** となった。

GPS 内蔵型携帯電話の測位精度は、陸上の固定点に比べて精度が劣化する。GPS 内蔵型携帯電話の精度が劣化した原因として、やはり船舶特有の構造からマルチパスの影響や、海面反射の影響を受けて精度が劣化したのではないかと考える。

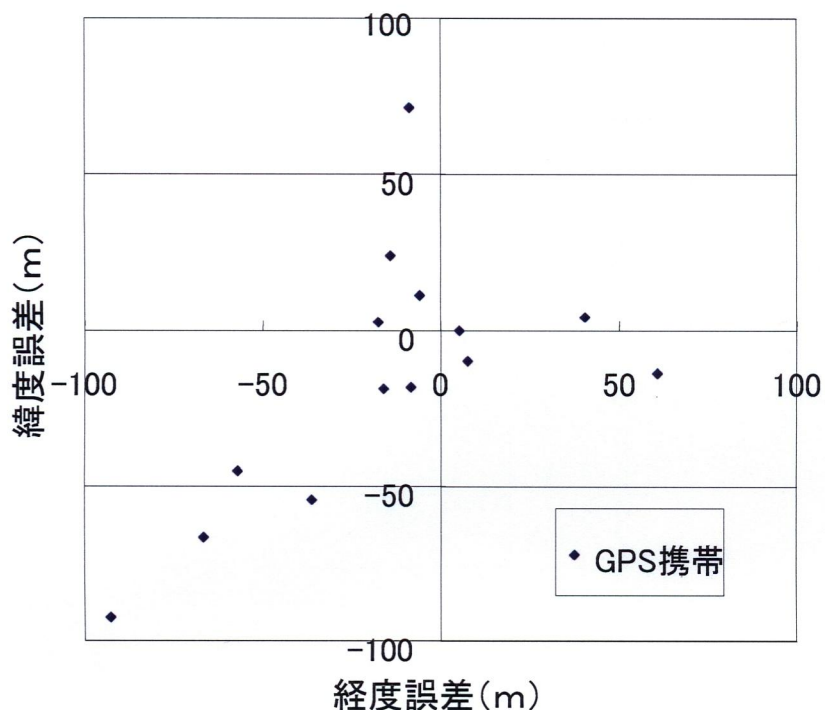


図 1.20 航海中の船橋における GPS 内蔵型携帯電話の測位精度

1.7.5 船内各所における測位精度

練習船汐路丸の船内各所、30箇所において測位実験を行った。測位ポイントから一番近い、窓までの距離と測位モードと誤差の関係を調査した。結果、図 1.21 の結果になった。緑色の丸では GPS 測位モードとなり、青いポイントでは、基地局を使った測位モードとなり、赤色の×の点では測位が不可能となった結果である。

図 1.22 に窓からの距離と誤差の関係を測位モード毎に表示した。この結果から窓から 4 m 付近までは、GPS 内蔵型携帯電話を使用して位置情報が取得できる。

また、表 1-4 の結果から窓に近い方が GPS 測位モードになることが多いといえる。

窓から 1m まででは、GPS 測位モードとなる確率が 66.7% となり、窓から 3m 以上離れると GPS 測位モードになる確率は 0% となり、基地局を使った測位モードの確立が高くなることがわかった。

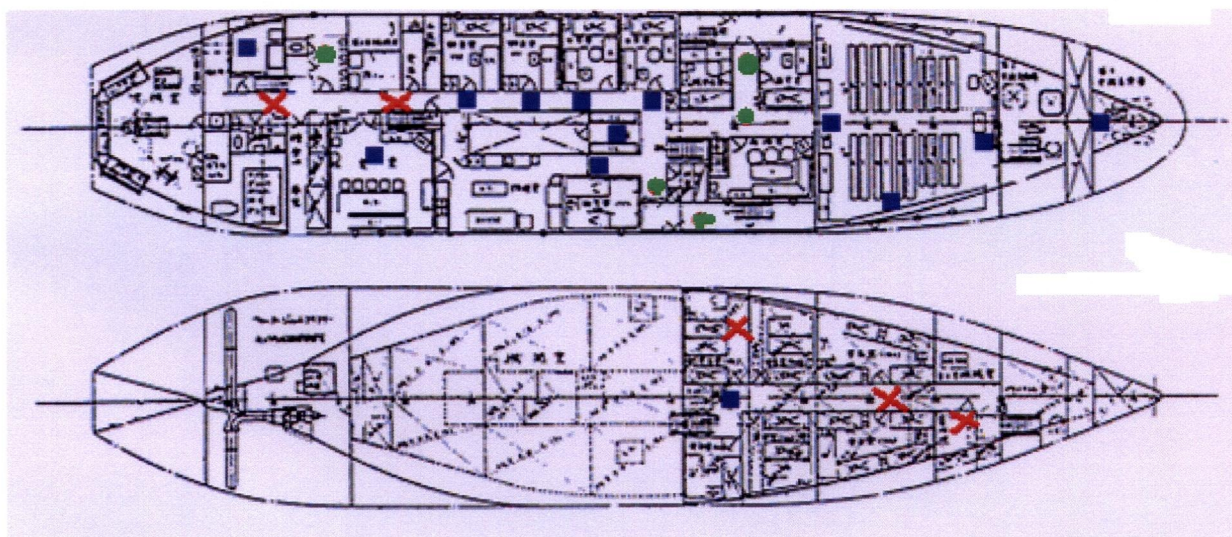
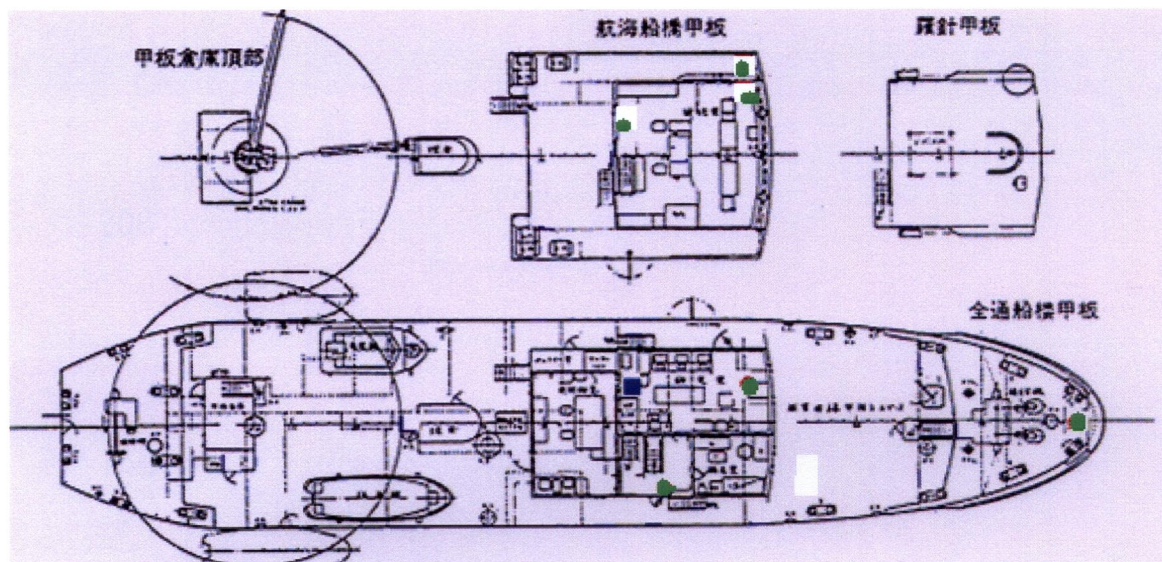


図 1.21 船内 30 箇所の測位実験

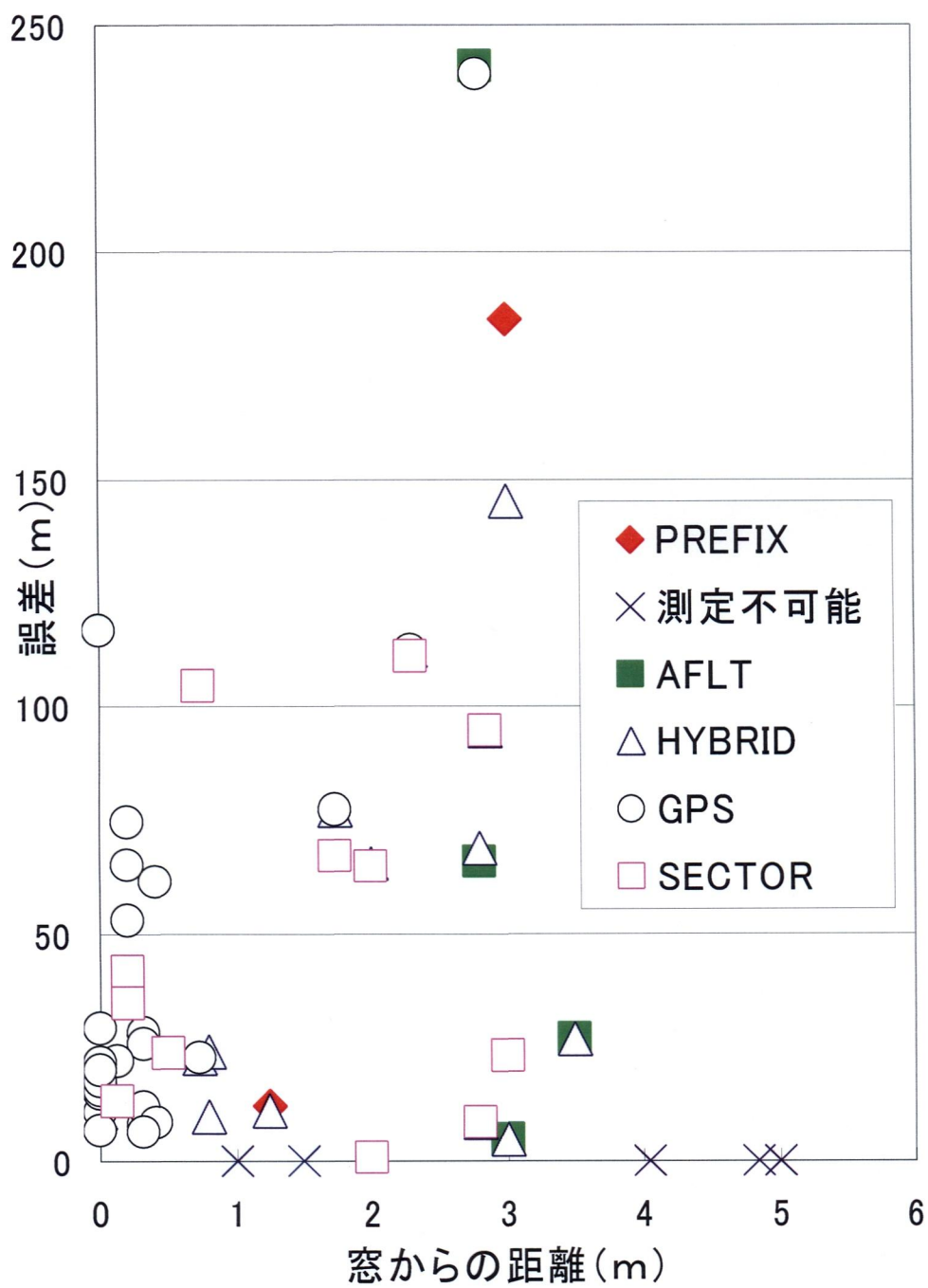


図 1.22 窓からの距離と測位モードと誤差の関係

表 1-4 窓からの距離と測位モードの変化

	GPS	基地局	測位不可能
1m以内	66.7	30	3.3
2m以内	8	76	16
3m以内	6.7	90	3.3
3m以上	0	68.6	31.4

1.7.6 勝どき停泊中における測位精度

東京海洋大学 練習船汐路丸が係留されている勝どきにて、GPS 内蔵型携帯電話の測位精度を Web 上にリアルタイムで見た(図 1.23)ものである。大体の位置は把握でき、GPS 内蔵型携帯電話の端末を持った、船員の位置が確認できる。



図 1.23 勝どき橋停泊中の船橋での測位結果

しかし、時々ではあるが、数十メートルずれている結果になった。図 1.24 に示す通りである。今回は連続測位、一度測位する際には、20 秒ほどかかるので、連続測位した結果で、

何度かデータを見て大体、ここにいるということが判断できるのではないだろうか。



図 1.24 勝どき橋停泊中の汐路丸の場所

1.7.7 陸上からの安全管理について

図 1.25 に示しているのは、船員の位置情報を Web 上で見たものである。これは、船内にいる船員に GPS 内蔵型携帯電話を持たせ、決めた時間に測位ボタンを押してもらうという実験をした。そして、陸上で、船員がどこにいるのか時系列で表示、記録したものである。この実験から、船員がこの端末を持ち測位ボタンを押すことによって自動的に陸上にいる人が監視することができる。これは、陸上からも船内でも位置情報を見ることができ船員の安全管理に使用できる。

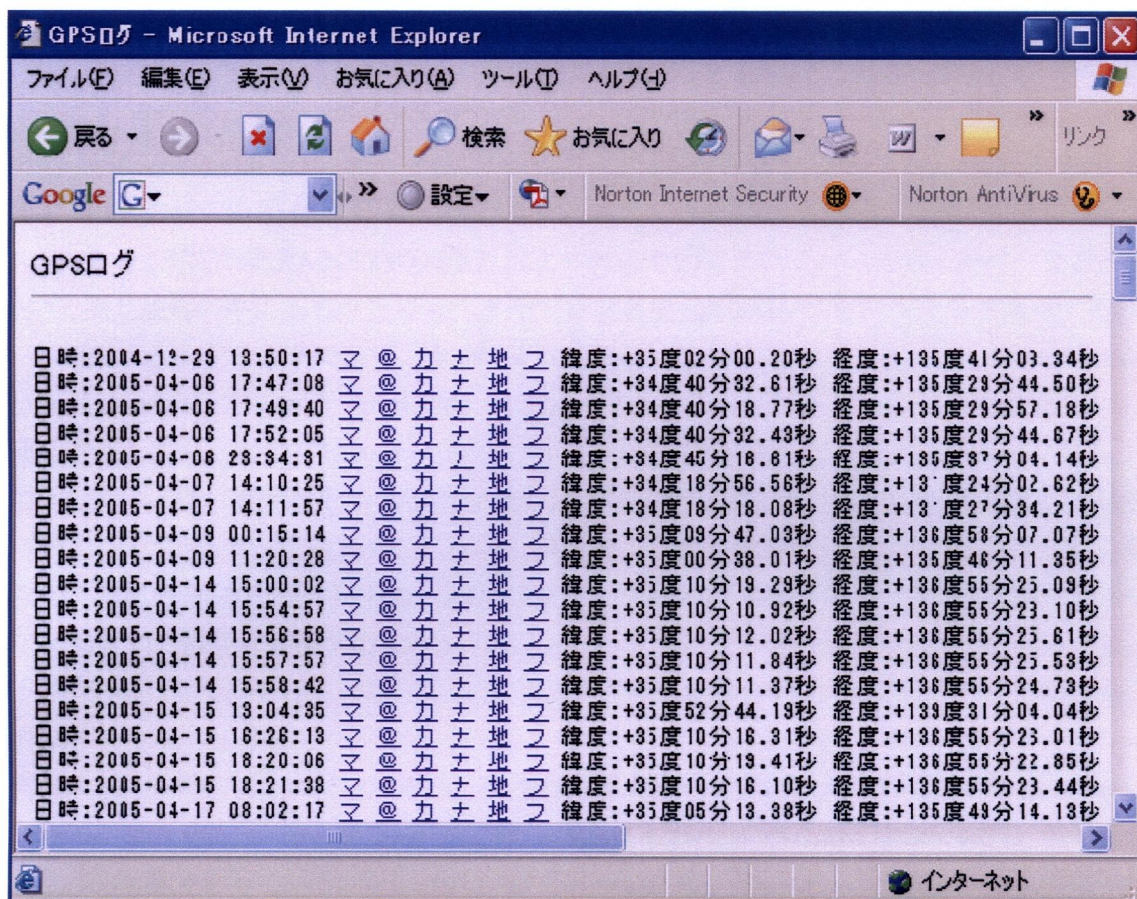


図 1.25 陸上からの安全管理システムについて

1.8 まとめ

現在、普及率の高い携帯電話の機能に GPS 受信機が内蔵されることを利用し、船員の位置把握や安全確認に使用できないかと考えた。今回、KDDI の携帯電話を使用した、通話や通信可能エリアが陸上から 20～30 マイルの範囲で利用できる。また GPS 受信機の機能の GPS 内蔵型携帯電話は、船内の窓際 4m までは位置情報を取得することが可能であることや、色々なアプリケーションを使用し、陸上での管理システムが容易に作成できることから船員がこの GPS 内蔵型携帯電話の端末を持つことで、船内においても、陸上からでも船員の位置情報が確認できるので、安全管理に使用できるといえる。

2 章

無線 LANVoIP

2.1 無線 LAN の概要

LAN (LOCAL AREA NETWORK) はより対線や同軸ケーブル、光ファイバーなどを使って、同じ建物の中にあるコンピュータやプリンタなどを接続し、データをやり取りする(データ通信) ネットワークである。通信制御方式によって Ethernet, FDDI(Fiber-Distributed Data Interface, 光ファイバーを利用したもの)、Token. Ring などのいくつかの種類がある。最も普及しているのは、Ethernet 規格でその中でも対線を使ったスター型 LAN である。10BASE-T (最大通信速度が 100Mbps,最大送信距離が 100m) と 100BASE-TX (最大通信速度が 100Mbps,最大伝送距離が 100m) が主流。対線とは電線を 2 本ずつ撚り合わせて対にした通信用ケーブル。平行型の電線に比べてノイズの影響を抑えることができる。各ペアの周りに、雑音を遮断するシールド加工を施したものを「STP (Shielded Twisted Pair) ケーブル」、シールドしていないものを「UTP (UnShielded Twisted Pair) ケーブル」という。

無線 LAN の規格については、現在、特殊な用途を除いて、ほとんどの LAN は Ethernet である。Ethernet は Xerox 社と DEC 社 (現在は Hewlett Packard 社の一部門) が考案した LAN 規格であって、IEEE(Institute of Electrical And Electronic Engineers:米国電気電子学会)の 802.3 委員会によって標準化されたもので Ethernet のことを IEEE802.3 とも呼ばれている。そして、IEEE の 802 委員会ワーキンググループ 11 が 1998 年 7 月無線 LAN の標準規格としていくつかの規格を定め、これらの規格群を IEEE802.11 と言う。この規格群にある無線 LAN の規格を表 2-1 にまとめる。

表 2-1 無線 LAN の規格

	IEEE 802.11b	IEEE 802.11a	IEEE 802.11g
周波数	2.4GHz帯	5.2GHz帯	2.4GHz帯
通信速度(規格値)	最大11Mbps	最大54Mbps	最大54Mbps
通信速度(実質値)	最大5Mbps	最大22Mbps	最大20Mbps
伝送距離	50m～180m	20m～90m	20m～80m
互換性	高い	なし	bより低い
電波干渉の可能性	Bluetooth/電子レンジ	なし	Bluetooth/電子レンジ
メリット	互換性が高い 安価 比較的安定している	bより高速 他の周波数帯より空いている 屋外で利用できる	bより高速 IEEE 802.11bと互換性がある 遮蔽物も透過性が高い
デメリット	速くはない Qosの欠如 WEPが脆弱 同一セルに3台のAPを置くと、 ローミングができなくなることがある	他の方式と互換性がない 電力消費が厳しい 遮蔽物も透過性が低い 高価	他の周波数帯より混んでいる 比較的不安定で、高価

本研究では、IEEE802.11b 規格の無線 LAN を使用した。IEEE802.11b では 2.4GHz の周波数を利用している。これは、ISM(Industry-Science-Medical)帯域と言われ、「産業、科学、医療機器、工場生産ラインなど」の無線機器において、自由に使用される周波数帯域である。使用に関しては、各国が独自に管理・認証を行っており、日本の場合は無線免許なしで利用できる。

変調にはスペクトラム拡散方式を採用し、1Mbps に DBPSK(Differential Binary Phase Shift Keying)、2Mbps に CCK(Complementary Code Keying)と呼ばれる搬送波の位相差を使った 1 次変調を行い、DSSS(Direct Sequence Spread Spectrum 直接シーケンス・スペクトラム拡散方式)の 2 次変調で 20MHz の帯域に拡散する。DSSS は信号に PN (疑似ノイズ) 符号系列を掛け合わせ信号を拡散する方式であり、高速通信に用いられる。この方式は拡散に使用した同一の信号を使用しなければ復元できず、隠匿性と干渉に強く、現在、多くの無線 LAN ベンダは DSSS 方式を採用している。これに対して、FHSS(Frequency-hopping spread-spectrum:周波数ホッピング方式)は搬送波の周波数を切り替えて通信する方式であり、狭帯域での搬送に適している。

無線 LAN (IEEE802.11b) によるネットワーク構成

アクセスポイントを使うか使わないかによって、無線 LAN によるネットワークには次の 2 つのタイプがある。まず、はじめにアドホック (ad-hoc) 又はピアツーピア (peer-to-peer)

方式：アドホックとは、最も基本的は無線 LAN 構築法であり、無線 LAN カードを実装した PC（無線 LAN 端末）間で無線通信を可能にする。一般的にこの通信方式は 1 対 1

（peer-to-peer）のアドホック・ネットワークと呼び、この通信方式ではネットワーク管理者の専任化あるいは格別のネットワーク構成なしで各無線 LAN 端末から他無線 LAN 端末へリソースのアクセスが行える。アドホック無線 LAN は小規模な部門内通信あるいは SOHO(Small Office/Home Office)の用途に適する。Ethernet では、1 つの通信経路を共有するために、SMA/CD(Carrier Sense Multiple Access/Collision Detection)というアクセス制御方式が用いられているが、無線 LAN では離れば弱まるという電波の性質上、送信中のノードが衝突検出を行うことはできない。そこで、受信側の ACK (Acknowledge:肯定応答) 応答の有無によって、衝突を回避する CSMA/CA6with ACK (Carrier Sense Multiple Access/Collision Avoidance with ACK : デジタル値をアナログ信号に変換する変調方式の一つで、正弦波の振幅に値を割り当てる方式) というアクセス制御方式が用いられている。また、無線 LAN 端末間において、相互の距離が届く範囲外にある端末による通信の割り込み（隠れ端末問題）を回避するために、送信に先駆けてこれからを送信することを告げる「RTS (送信要求)」と、それに応じる「CTS(送信可)」を送り合う RTS/CTS(Request to Send/Clear to Send)というアクセス方法が用意されている。

もう一つの方式は、インフラストラクチャ (Infrastructure Mode) 方式：アドホックモードでは、互いのノードが電波到達範囲内に存在することを前提として通信が確立する。全ての端末が相互に通信するなら、サービスエリアの範囲内に全てを配置しなければならず、無線 LAN の運用は大変制約の多いものとなる。そこで、携帯電話や PHS の基地局と同じようにアクセスポイントを配置し、アクセスポイント間を、Ethernet 等を使って結ぶ運用形態が用意された。これをインフラストラクチャモードという。アクセスポイントは通常、サービスエリア内の端末が外部と通信できるように内外の中継を行うブリッジとして機能する場合が多く、また携帯電話や PHS と同様に、機器の仕様によっては稼働中の端末が各アクセスポイントのサービスエリア間を渡り歩く、ローミング（無線搭載の携帯コンピュータの利用ユーザが 1 つのアクセスポイントの適用範囲外に移動しても通信状態の一番良好な他のアクセスポイントに自動的に切り替え通信を可能とする機能）も可能な仕組みとなっている。

2.2 VoIP(Voice over Internet Protocol)

2.2.1 VoIP の概要

VoIP とは Voice over Internet Protocol の略で、IP ネットワーク経由で音声情報を送受信するための技術のことで、企業間では急速に普及している。

VoIP とは音声情報をパケットにして IP ネットワーク経由で送受信する技術である。電話網の構築・運用にかかるコストを削減できる技術として着目されていることに加え、既存の電話ではできなかったような例えば相手が無人でも遠方の場所の物音を聞いたり(監視目的)、IP パケットでうけとることから発信元の認証が容易にできたりするような応用技術も最近では注目されつつある。

また一方で、VoIP については、「音質面で大丈夫か」といった先入観念で見られている。もちろん VoIP が現在の電話を淘汰してしまうようなものではないにしろ、VoIP の弱点とメリットも認識した上でその機能を考えてみる。

大きなメリットのひとつであるコスト削減は、従来の電話網を経由せず IP ネットワーク経由で音声を送信することから実現するが、同時に IP ネットワークが一般的には、ベストエフォート型で QoS (Quality of Service: ネットワーク上で、ある特定の通信のための帯域と通信速度を保証する技術) 保証がないことが弱点であり、これを克服しいかに通話品質を確保し問題を解決していくことが、VoIP のシステム構築の際の課題であるとされている。

従来の電話網による音声通話が、ほぼ均一な条件下での利用であるのに比べて、VoIP では、ネットワークの規模、回線の品質、使用する VoIP 製品、ルータなどの機器により、その音声品質が大きく左右される。当然さまざまな規模、回線品質、製品があるので、VoIP を用いてコスト削減をはかりつつ業務上、既存の電話に比べて音質面でほぼ遜色なく通話品質を維持するためには、現状ほぼ最適と考えられる組み合わせを考慮して設計することで、これは VoIP も他のネットワーク機器と同様である。

データ通信との違い

ただ、データのみ通信と異なるのは、最後でデータが音声信号に変換され最終的に人間の耳に到達されるため、その品質評価は機械ではなく人間が行うことである。ここにもうひとつの VoIP の大きな特徴があり、またさまざまな品質向上のための工夫がなされている。たとえば、データ通信のみであれば途中でそのデータの一部が欠けた場合(パケットロス)には全く用をなさないケースがあるが、VoIP による音声情報伝達の場合は、人間の耳で確認するため多少のロスが発生してもそれを会話として認識できる程度で直前の音を再生する機能をもたせたりすることができる。(これは音声のみならず画像の伝送にも似たような技術ある) また従来のアナログ電話の技術であるエコーキャンセラ(相手に送った音声信号の一部が自分に戻ってくることをエコーという。これにフィルタをかけて除去

する技術)も音質向上に不可欠である。世の中には、多くの VoIP 製品がでまわっているが、このエコーキャンセラ機能を装備されたものを推奨されている。エコーを除去するエコーサプレッサ (echo suppressor) というものもあるが、これは簡便法 (会話が交互になされることを前提としており、戻ってくる送信信号を減衰して除去する方法) で両方が同時に話すいわゆるダブルトークの場合には両方の信号が減衰されて非常に聞き取りづらい状況が発生する。特に会話の最初や、遅延がある場合には両者が同時に話すことも実際には、時々ある。

VoIP では、ネットワーク経由で音声が行きわたるため、やはりネットワークのゆらぎや遅延などによる影響は、受けるためそれを軽減させるように努めているが、その程度によって音質は幅がある。ネットワーク回線の品質も、日によりまた時間帯により相当帯域などが幅のある環境下で、多くの音声通話を実現されているケースなどには、相対的な音質評価だなどの認識でいたほうがいいだろう。VoIP 構築に際しては、擬似環境でぜひデモ機を借りて自身の耳で評価することが推奨されている。音質は人間の聴覚が直接知覚するものだけに、きちんと数値化するのはなかなか難しい。

コスト削減の視点からいえば、現在の従量課金の電話網からなるべく多くの回線を VoIP 化することで、最大限の費用効果を実現したいところだが、現実には、大規模システムであればあるほど音声以外のデータとの統合も配慮することも多く、また拠点間の回線速度も様々なケースも多い。中継間はもとより自社内のトラフィックにも依存するが、ルータなど他の機器にも配慮が必要になってくる。これは VoIP に限らず、ネットワーク設計の際にも 拠点数が多くなればなるほど、通過する機器や回線の種類が増え、障害時の切り分けのための考慮すべき要因が増えることとほぼ同じである。多くの拠点間を結ぶ VoIP 網構築で回線速度も異なり、データも混在しているネットワーク環境では、熟練した方でも相当ハードルが高いと思われる。通話品質確保には相当安定した高速ネットワークが前提となる。

優先制御機能

特定のデータを優先的に送ってくれる優先制御機能がある。優先ルーティングとも呼ばれる。この機能を使うと特定の UDP ポート番号のデータを優先的に通してくれますので音声データの遅延や揺らぎを押さええることができる。とりあえず 2 拠点間で VoIP のための専用回線で構築していて、まずはここからスタートということで、障害も発生せず順調に稼働している。

音声とデータをどのように変換するのか

基本的に送信側では、音声信号を符号化圧縮（コーデック）によりデジタル処理をしてデータにし、パケット化した上で UDP/IP ヘッダなどの宛先情報を付け、適当な間隔（タイミング）で IP ネットワーク上に送出する。相手先では、その逆で 途中のネットワーク内部でのゆらぎ（ネットワーク上でのトラフィック状況によりパケットの到達間隔にばらつきができること）を吸収するためのバッファ（次にくるパケットを待つ）間隔をとり、複合化し、音声信号に再変換している。TCP はネットワークレベルで再送をしてくれるので便利なプロトコルなのだが、音声を転送する場合には VoIP は一般的に UDP/IP の組み合わせで使っている。TCP を使うと再送手順をネットワークに委ねることになる。

TCP というプロトコルではパケットロストが発生すると送信側タイムアウトするまで再送する仕組みの為、1つのパケットロストで場合によっては数百ミリ秒から数秒の遅延が発生してしまい会話にならなくなる。音声の場合にはパケットロストが発生しても再送するのではなく音声をごまかす処理を行うことによりリアルタイム性を優先させる方が好ましいとされている。また TCP に比べ UDP の方がパケットサイズが小さく、細い回線に多くの通話チャンネルを持たせようとした場合に有利になる。

コーデック処理に関しては、H.323（ITU-T 国際電気通信連合の勧告で Qos 保証のない IP ネットワーク上での音声、映像、データを通信するためのプロトコル体系）の G シリーズ 例えば G.723.1(6.3Kbps)または G.729(8Kbps)といった音声圧縮が VoIP では使われていることが多い。音声情報の大きさは一般の電話機の場合 G.711 で 64Kbps、VoIP の場合には通常音声を圧縮して利用する。現在 VoIP で主流になっている音声圧縮は G.723.1(6.3Kbps)または G.729(8Kbps)である。G.723.1 を例にとった場合 30 ミリ秒毎（= 秒間 33 パケット）に 24 バイトで、もしこの音声情報のみを送るとすれば $24\text{byte} \times 8\text{bit}/30\text{mS} = 6.3\text{Kbps}$ となるので音声情報を相手まで送るためには UDP/IP のヘッダーを付けなければならない。さらに Ethernet にのせるためには MAC アドレスの入ったヘッダが、専用線を通るときには PPP の為のヘッダが追加される。帯域を確保するための RSVP(Resource Reservation Protocol)を利用する場合にはさらにこの為のヘッダが追加される。

通信事業者向けの VoIP 製品の場合には比較的太いネットワークを利用することができるので問題にはならない。またネットワークの全てが 10Mbps や 100Mbps の LAN であればこの 10Kbps の情報量は問題になることは少ない。

このような用途では帯域よりも音質が優先される。しかしながら一個所でも細い回線を通るようであればできるだけ情報量を押さえて回線負荷を減らすことが望まれ、中小規模の内線網として利用する場合にはできるだけネットワーク維持費を押さえなければなりませんので細い回線でも利用できるような VoIP にすることが望まれている。

ソフトウェア処理とハードウェア処理の違い

ソフトウェア処理をしている VoIP 製品は、専用の DSP ボードを持たず、音声圧縮、伸長処理をすべてパソコンなどの CPUで行っている。いくら CPUが高速でも他のドライバソフトが一定期間 CPUを開放しないようなシステムでは不安定になり、リアルタイム処理ができないために音声の圧縮伸長にも限界がある。

また、これらのシステムは2線式の通常の電話機を接続できない。現在運用中の電話システムのコアである PBX や公衆網と相互乗入れすることが困難でありこれらの VoIP を企業向けに利用することは困難である。その理由としてはハードウェア上の理由以外にエコーキャンセラ処理（相手に送った音声信号の一部が自分に戻ってくることをエコーという。これにフィルタをかけて除去する技術）をリアルタイムに処理することが難しいことが挙げられる。図 2.1 に VoIP のシステム図を示す。

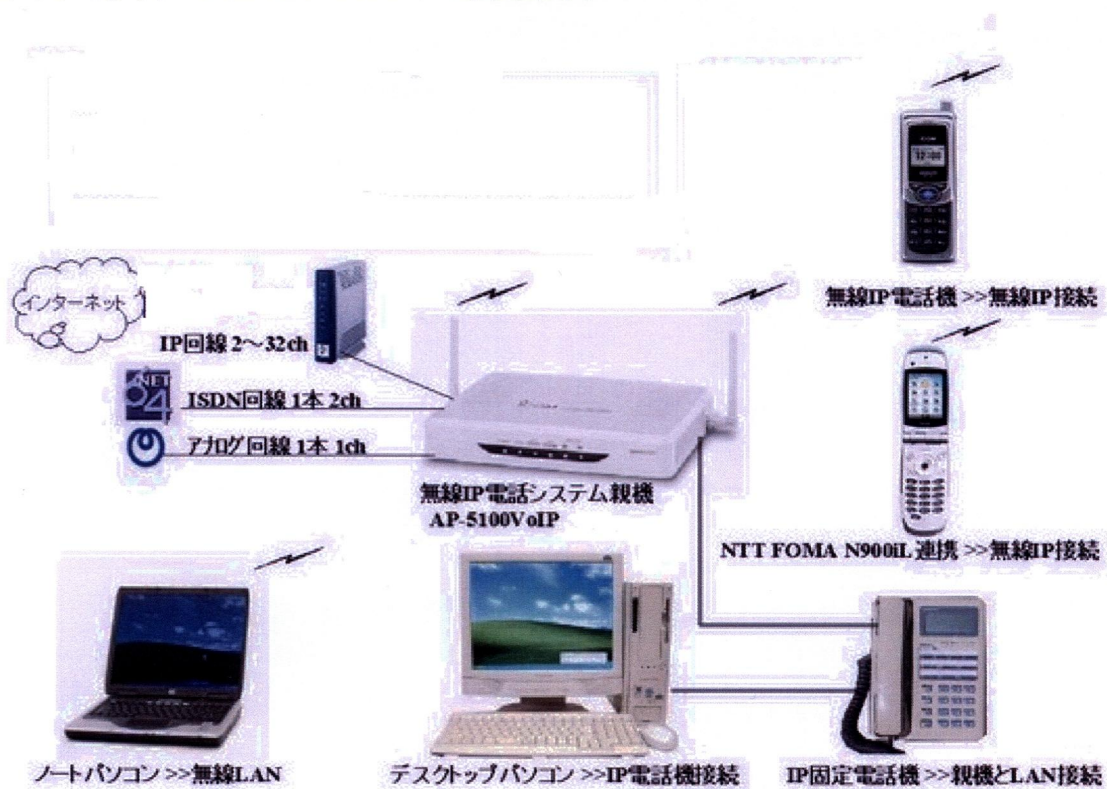


図 2.1 VoIP のシステム図

2.2.2 IP 電話システム内で活用する携帯型無線 IP 電話

今回、実験に使用したのはワイヤレス IP フォン(携帯型)で VP-43 型と言われるもので、IP 電話システム内で活用する携帯型無線 IP 電話である。特徴として、IP 電話システム内の無線子機として外線や内線通話に対応している。802.11 無線 LAN に対応していて、暗号化セキュリティは WEP を搭載していて、Web ブラウザから本体設定が可能で、小型で軽量である。図 2.2 は今回、実験に使用した VoIP(VP-43)とアクセスポイント兼ルータである。アイコム(Siigam)の SR-5200VoIP はブロードバンド無線ルータで VoIP 対応で VoIP 対応というと単なる ATA 機能が搭載されているものが多い中、この SR-5200 は FXO ゲートウェイとしても動作する本格的なもので、さらに SIP 簡易サーバ機能が搭載されているため、1 台あれば家や小規模なオフィスの中をすべて IP 電話化することまでできてしまうという優れた点を持っている。相当数の IP 電話/ソフトが接続可能で、Asterisk の FOX 代わりと 050 番号の IP 電話ゲートウェイにも使用できるので、応用範囲の広い製品である。



図 2.2 実験に使用した WVoIP 端末とアクセスポイント

2.2.3 NetStumbler を用いたネットワークの測定

2.2.3.1 NetStumbler の概要

NetStumbler は信号強度を測定し、複数のネットワークが同じチャンネルを使用している場合には、信号強度の表示を得られる。NetStumbler(<http://www.stumbler.net/>)はフリーのソフトで、到達範囲内にあるすべての無線ネットワークに関する多量の詳細情報、ESSID(Extended Service Set Identifier:IEEE 802.11 シリーズの無線 LAN におけるネットワークの識別子の一つで、混信を避けるために付けられるネットワーク名のようなもの。最大 32 文字までの英数字を任意に設定できる。アクセスポイントの識別子だった SSID を、複数のアクセスポイントを設置したネットワークでも使えるよう拡張したも

の。),WEP(Wired Equivalent Privacy:無線通信における暗号化技術。無線通信は傍受が極めて容易であるため、送信されるパケットを暗号化して傍受者に内容を知られないようにすることで、有線通信と同様の安全性を持たせようとしている。)を使用しているかどうか、使用しているチャンネル等を得ることが可能である。NetStumbler の最も優れた機能の一つは、信号強度を MIDI を用いたフィードバックで提供している点で、「長距離リンクのためのアンテナ調整法」で試すように、2 点間での最良の信号を発見する際に便利である。信号強度が増加するにつれ、NetStumbler が出すトーンのピッチが上がり、これによりパラボラアンテナを向けるのと同じように、アンテナを調整することができる。つまり、最も高いピッチのトーンが聞こえるまで、アンテナの位置を変えることができる。NetStumbler は色でアクティブなリンクを表示し、緑は強い信号、黄は普通、赤はほとんど使えない場合、灰色は無線ネットワークが圏内にない場合を示す。

NetStumbler が発見したすべての無線ネットワークの信号強度、SN 比、ノイズがわかる。非常に便利な視覚化ツールで、指向性のあるアンテナを適切に設置する場合にも、このツールは有効である。この機能は、企業のオフィス環境でアクセスポイントの最良の配置場所を決定する際にも利用される。

2.2.3.2 船内における VoIP の通話実験

東京海洋大学練習船、汐路丸の船内において、VoIP の通話実験を行った。そこで、ネットワークの検出という点から NetStumbler での検出を行い、どの程度の信号強度で通話できなくなるか。また汐路丸の大きさの船舶には、ルータ兼アクセスポイントは何台いるのだろうかという点を検証する為に実験を行った。図 2.4 は汐路丸の構造である。

図 2.5 は汐路丸で、通話実験を行った結果、アクセスポイントは各甲板に 1 つ、見通し距離が悪い場合は、2 つ設置すると船内全体で通話が可能であることがわかった。また、NetStumbler を使用し、SNR ((Signal-to-Noise ratio):有効な信号成分に対するノイズ成分の割合で、値が大きいほどノイズの少ない良好な信号であることを示す) は、30dB 以下になると通話することができないことがわかった。

航海船橋甲板と羅針盤甲板において音声通話実験を行い、信号強度を測定した場所を図 2.3 に示す。各場所で測定した結果を表 2-2 に示す。この結果から、各甲板以外に通話できる場合も 30dB 以下(Network Stumbler 計測)になると通話ができなることがわかった。また、アクセスポイントの設置状況によって、上の甲板まで電波が届き通話が可能になることがわかった。

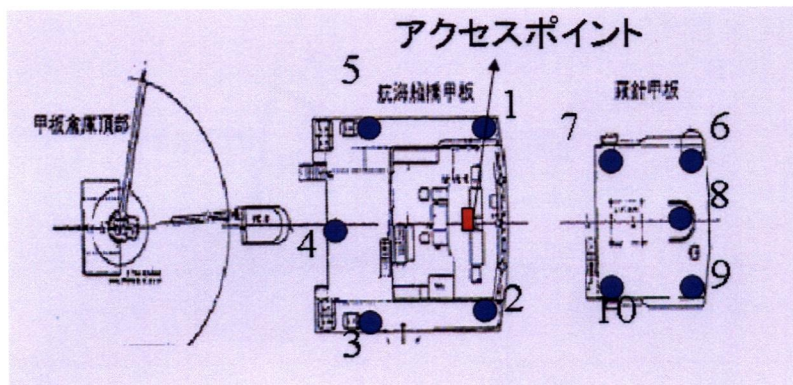


図 2.3 信号強度を測定した場所

表 2-2 信号雑音比の測定結果

測位した場所	1	2	3	4	5	6	7	8	9	10
SNR(dB)	61	50	41	41	49	41	24	21	23	21

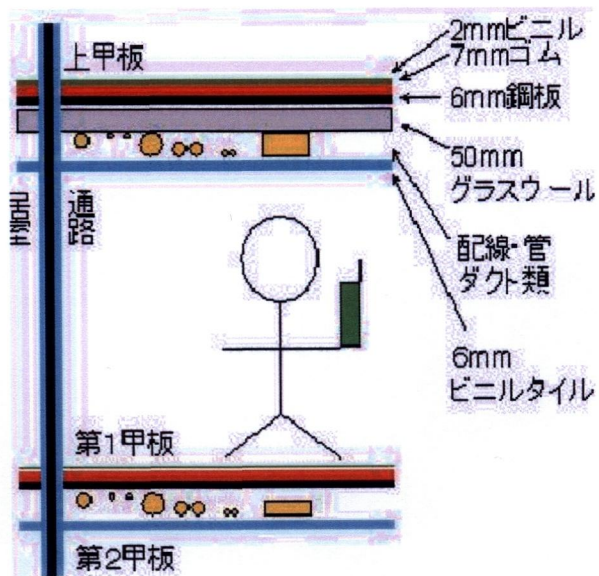


図 2.4 汐路丸の甲板の構造

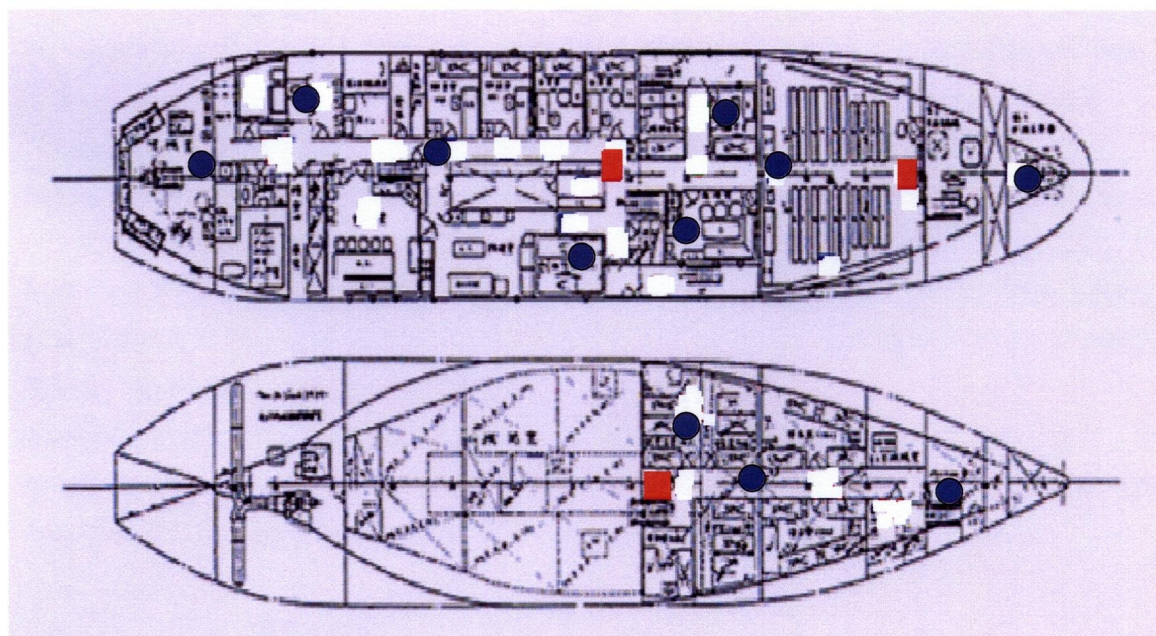
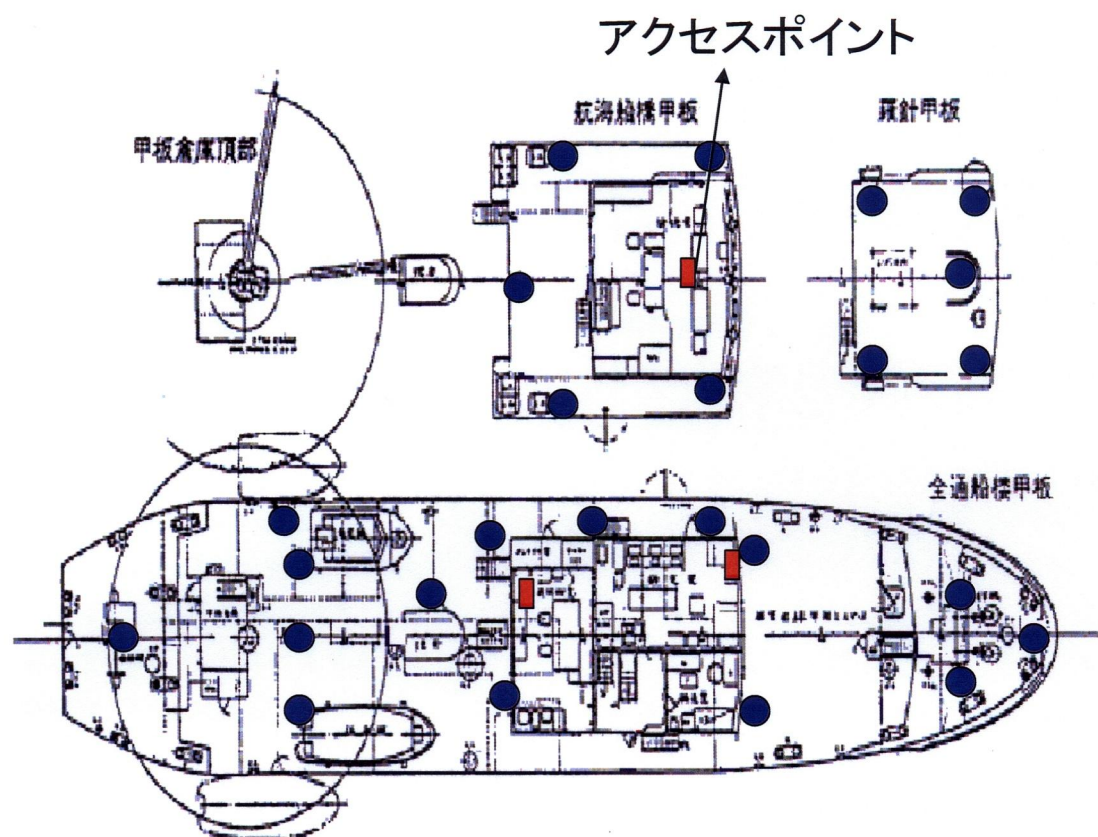


図 2.5 通話実験から汐路丸の場合のアクセスポイント設置場所

2.2.4 Asterisk とは

オープンソースの IP-PBX ソフト「Asterisk」が注目されている。欧米では、「Asterisk」(アスタリスク)を使った IP 電話システムの利用が始まっている。

無料とはいえ機能はメーカー製の IP-PBX ができることはほとんど可能で、IP 対応 PBX、従来 PBX、対話型音声応答 (IVR) システム、VoIP ゲートウェイ、電話会議などさまざまな電話ソリューションの構築に利用されている。

「Asterisk」は単なる IP 電話サーバではなく、着信転送や不在転送など PBX 機能を持っている。固定電話とも接続が可能。ボイスメールの機能設定の簡略化動的 IVR (音声自動応答システム) フロー管理インターフェースの刷新 PRI (1 次群インターフェイス) の改良 SIP プロトコルサポートの改良 CDR(通話記録)のサポートリアルタイムのデータベース設定など、こうした機能を既存の PBX や SIP サーバと組み合わせて利用することができる。

「Asterisk」では、複数の呼制御プロトコルをサポートし、H.323・MGCP・SCCP と言った従来からあるものに加え、Asterisk 独自のプロトコルとして LAX2 というプロトコル実装している。すでにアメリカでは LAX2 を実装した IP 電話が登場している。また、多くの呼制御プロトコルをサポートしているので、プロトコル変換アプリとして使うことも可能である。

日本国内でも電話の利用のあり方が大きく変化しており、従来は、電話は高価なものであり、企業では部署ごとに 1 台置いて、電話があれば受話器をとった人が通話相手を呼び出すという使い方であったが、従来の電話に加え、ソフトフォンやオープンソース PBX

「Asterisk」を利用すれば、業務に合わせて電話網を導入することが可能である。既に、Skype を始めとするソフトフォンでは実現できているが、Web ページやエクセルなどのシートに発信機能を貼り付け、そこから通話することもオープンソース PBX「Asterisk」を利用すれば可能 Asterisk そのものの登場はまだ新しく、2004 年 9 月 23 日に開発された。日本では個人で PBX に注目するユーザがさほど多くないため、日本語のドキュメントはあまりないのが現状である。

Asterisk は、フルセットの PBX として考えられるすべての機能が詰まっている。また、IP 電話機能にかんしては、主要な 3 つのプロトコルをサポートしていて、各種の IP 電話機能と連動動作することができる。ソフトウェアベースの PBX である。

Asterisk の構造には次の図のようになる。

Asterisk のコアに相当する部分には以下のものがある。

- ・ PBX Switching : 基本的なソフトスイッチの部分で、PBX 機能そのものである。
- ・ Application Launcher : 各種サービスを提供するプログラム、たとえばボイスメールなどのアプリケーションを起動する。

- Codec Translator : 各種コーデックの相互変換を行う
- Scheduler and I/O Manager : 低レベルのスケジューリング処理と動作の最適化を行う
このコア部分以外の機能は大半がロードابل（動的組み込み）モジュール構造になっている。
- Channel Interface : 各種の接続を行う部分で SIP は H.323 の接続はこの部分が担当する。
Asterisk Channel と呼ばれている。従来の電話回線も IP 電話もすべてこのチャンネルとして扱われる。
- Manager Interface : PBX の管理インタフェースを提供する部分
- AGI : AGI とは Asterisk Gateway Interface の略で、CGI のように Asterisk から外部プログラムの実行をサポートする。

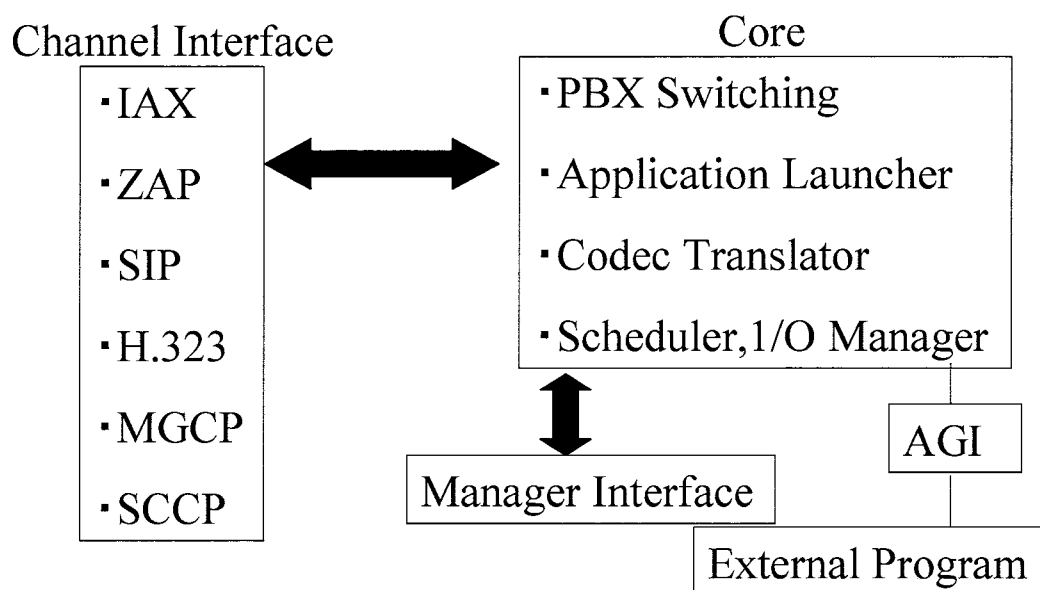


図 2.3 Asterisk の構造

2.2.5 SIP サーバを使用した実験

SIP (Session Initiation Protocol)

VoIP を応用したインターネット電話などで用いられる通話制御プロトコルの一つである。1993 年 3 月に発表された規格で H323 など同様のプロトコルにより後発のため、まだあまり普及はしていない。各端末に割り当てられるアドレス形式が電子メールアドレスの形式に近く、将来的には共通化も可能とされている。

この Asterisk 機能の SIP サーバを使用し、VoIP 端末からの登録手続きを行った。

そして、IP 電話の特徴ともいえる、IP アドレス (IP Address) に着目し、IP アドレスは確

認信号が個別に認識できるという特徴を持っている。そこで、arp その端末の固有の ID がわかる。arp とは Address Resolution Protocol の略でテーブルの表示や設定を行う。arp テーブルとはインターネット通信のために用いられている。そこで、この標準コマンドを組み合わせれば、船内の VoIP 端末の ON,OFF やある程度の場所（アクセスポイント）のそばにいたことがわかるのではないかと考えた。図 2.6 に VoIP 端末を使った位置把握システムの仕組みを示す。

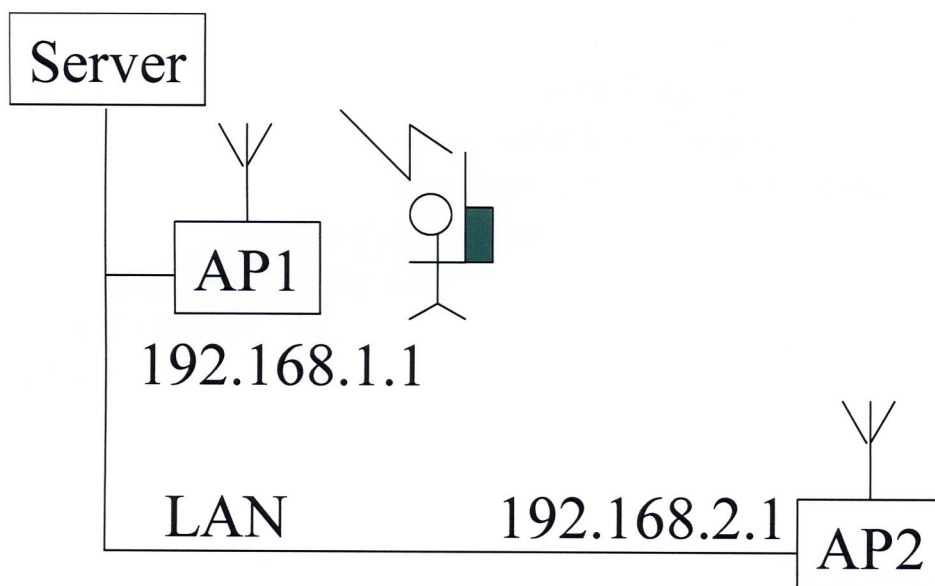


図 2.6 VoIP を使った位置把握システムの仕組み

船内において、2 人の被験者が、VoIP 端末(VP-43)をの電源を入れて船内を歩行した時の様子を Asterisk サーバを使って、30 秒毎にデータを記録したものを図 2.7 に示す。

time	端末番号
Jul 25 14:29:56 chan_sip.c: Auto destroying call *****@192.168.1.102' →AP	
Jul 25 14:29:58 chan_sip.c: Auto destroying call *****@192.168.1.10' →VP-43	
Jul 25 14:29:59 chan_sip.c: Auto destroying call *****@192.168.1.11' →VP-43	
Jul 25 14:30:28 chan_sip.c: Auto destroying call *****@192.168.1.10'	
Jul 25 14:30:29 chan_sip.c: Auto destroying call *****@192.168.1.11'	
Jul 25 14:30:56 chan_sip.c: Auto destroying call *****@192.168.1.102'	
Jul 25 14:30:58 chan_sip.c: Auto destroying call *****@192.168.1.10'	
Jul 25 14:31:00 chan_sip.c: Auto destroying call *****@192.168.1.11'	

図 2.7 Asterisk サーバで端末の状態を記録したもの

図 2.5 に示すように、アクセスポイントを 192.168.1.102 の IP 番号にし、各 VoIP 端末 (VP-43) を 192.168.1.10 と 192.168.1.11 の IP 番号を付けたものである。この結果からこの端末を持った人は、192.168.1.102 番のアクセスポイント付近にいることが判断できる。これは、Asterisk 機能の SIP サーバの登録手続きを端末から 30 秒毎に自動的に行う設定にしているため、どこのアクセスポイントを使って登録手続きが行われているかが判断できる仕組みになっている。この機能を利用することで、船内のどこにいても船員がこの端末を持っていれば、どこのアクセスポイント付近にいるかが判断できる。

2.2.6 まとめ

船内で VoIP 端末を使用し、船内のパーソナルコミュニケーションと船員の位置把握として使用できないかと考えた結果、船内におけるパーソナルコミュニケーション手段として VoIP は使用が可能であることがわかった。また、VoIP の電話交換機機能を持った Asterisk 機能を使用することによって、端末の状態を管理することができ、この端末を持った船員がアクセスポイントの番号を工夫することで、どこのアクセスポイント付近にいるかが判断できる。図 2.8 は、汐路丸での場合の VoIP を使用した位置管理システムのネットワーク構成案である。

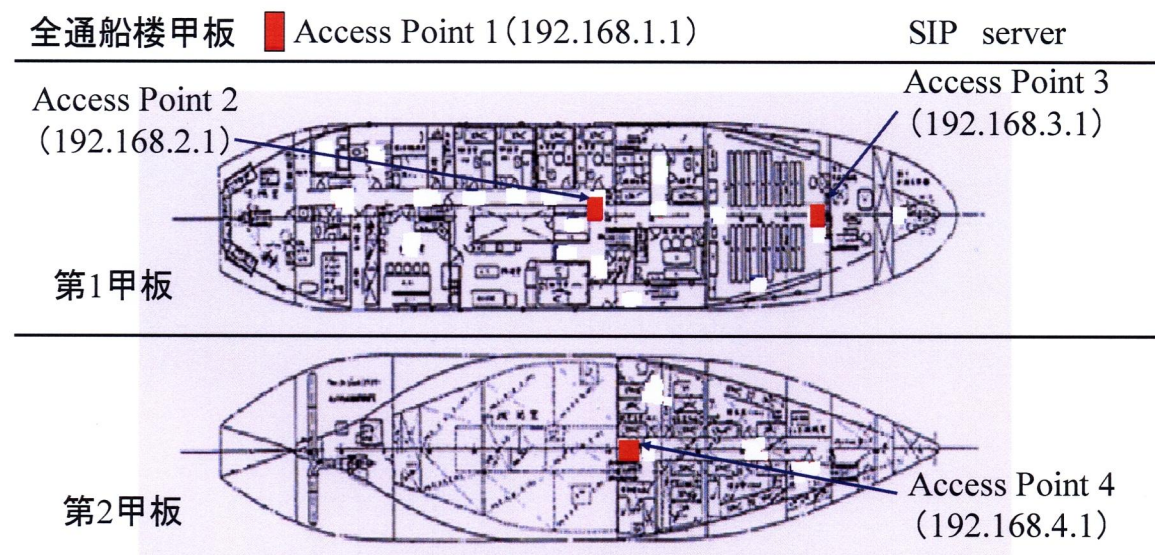


図 2.8 汐路丸の場合でのネットワーク構成案

3 章

結論

将来、携帯電話に電話機能の他に GPS 受信機と無線 LAN VoIP が搭載されるという点から、船員のパーソナルコミュニケーションと私用通信と安全管理に使えないかということで研究を行った。そこで、GPS 内蔵型携帯電話は船内では窓際 4m までは測位ができ、陸から 20～30 マイルの通信可能範囲内であれば、船員が取得した位置情報を陸上でも監視、管理できるということが分かった。また、船内のどこでも船員のコミュニケーションや安全管理ができるように、無線 LANVoIP を使用し、船内での連絡等に利用できるか実験を行った結果、船内ではどこにいてもアクセスポイントを置くことで使用可能であった。

また、無線 LANVoIP 端末の通信の可否等を Asterisk で記録することによって、この端末を持った船員の安全が確認できると考えられる。また、IP アドレスを工夫することで、どここのアクセスポイント付近にいるかが判断できる。将来、携帯電話に携帯電話機能の他に GPS 受信機と無線 LANVoIP が内蔵され、この携帯電話を持った船員は、GPS 内蔵型携帯電話により絶対的な位置と、無線 LANVoIP をを持った船員の大まかな位置が判断できる。この携帯電話端末を船員が持つことによって、安全確認が可能になる。

今後、陸船間通信の充実化によって、船員個人の携帯への私用電話、記録、支払いから安全管理まで応用的な利用が見込まれる。

参考文献

1.IP ネットワークを利用した航海データ収集・転送システムの開発
・練習船深江丸における実装・

若林伸和、矢野吉治、林祐司、村井康二

2001 年日本航海学会論文集 106 号 p29-37

2.海上無線 LAN を利用した小型船舶対象の安心ネットワーク構築
浦上美佐子、松野浩嗣、岩崎寛希

2004 年日本航海学会論文集 111 号 p173-180

3.無線 LAN による船内 LAN 構築と船内監視システム
三輪誠、若林伸和、矢野吉治

2005 年日本航海学会論文集 113 号 p243-250

4.PHS 無線 LAN を用いた海中転落者探知システム
鈴木治、田内大仁、橋本健一、浪江宏宗、安田明生

2000 年電子情報通信学会総合大会論文集 B-1-30

5.平成 18 年度の海難及び人身事故の発生と救助状況について
海上保安庁

6.GPS 測量の基礎

土屋淳、辻宏道 日本測量協会

7.GPS 人工衛星による精密測位システム

日本測位学会 編著 日本測量協会

8.Asterisk でつくる IP 電話システム

高橋隆雄著 秀和システム

謝辞

本研究を進めるにあたり、様々な形でご指導頂いた、東京海洋大学情報通信工学研究室の安田明生教授、久保信明先生に深く感謝申し上げます。

また、研究の動機や実験方法の案なども提供していただいた鳥羽商船高等専門学校の鈴木治先生に心から感謝致します。ありがとうございました。

海上実験の際、実験方法の助言、機材の収集から実験の手伝いまでお世話になりました川村雅志技官に深く御礼を申し上げます。

実験の手法や GPS の基礎勉強を教えていただいた防衛省防衛大学校の浪江宏宗先生に感謝しております。ありがとうございました。

研究の方向性や実験の仕方等をご助言いただきました、客員教授の北條晴正先生、東京海洋大学の樊 春明先生、横河電気の新井貴之さんありがとうございました。

研究の手法や発表の仕方等様々な形でご指導頂いた、サレジオ工業高等専門学校の吉田将司先生に感謝の意を表します。

また、船員という立場で色々な意見や実験の手法を教えていただいた独立行政法人 航海訓練所の冨永敏明助教授に御礼申し上げます。

本研究を行うにあたり、ご尽力いただいた本学の練習船汐路丸木村船長をはじめ乗組員の皆様に御礼申し上げます。

汐路丸の実験データを提供していただいた本学の田丸人意助教授に感謝の意を表します。

汐路丸の実験の際に、WVoIP の実験に協力していただいた情報通信工学研究室の土本和彦さん、湯浅純一さんありがとうございました。

汐路丸実験の際に GPS 内蔵型携帯電話のデータ取得に協力していただいた日本工業大学の桑名淳さんに謝意を表します。

本研究は船井電機㈱との共同研究の一部であることを記し、同社からのご支援に対し心より謝意を表します。

本研究にあたって様々な形でご協力して頂きました、既に卒業された方を含め情報通信工学研究室の皆様に厚く御礼申し上げます。

そして、学資の援助等、影ながら支えてくれた両親に深く感謝の意を表します。

付録

GPS 内蔵型携帯電話のデータ取得方法

MsGpsLocationTest.java

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class MsGpsLocationTest extends MIDlet{
    LocationCanvas canvas;
    public MsGpsLocationTest(){
        canvas = new LocationCanvas(this);
        Display.getDisplay(this).setCurrent(canvas);
    }
    public void startApp()throws MIDletStateChangeException{}
    public void pauseApp(){}
    protected void destroyApp(boolean unconditional)
        throws MIDletStateChangeException {
    }
    void doExit() {
        try {
            destroyApp(false);
            notifyDestroyed();
        }catch(MIDletStateChangeException e) {
        }
    }
}
```

LocationCanvas.java

```
import javax.microedition.lcdui.*;
import com.kddi.system.*;
```

```

class      LocationCanvas      extends      Canvas      implements
CommandListener,LocationListener{

    private MsGpsLocationTest main;

    private GpsLocation location;

    private String datum="未取得";
    private String unit="未取得";
    private String lon="未取得";
    private String lat="未取得";

    private String version="未取得";
    private String altitude="未取得";
    private String time="未取得";
    private String fixMode="未取得";

    private String majorAxisAngle="未取得";
    private String verticalError="未取得";
    private String semiminorAxisError="未取得";
    private String semimajorAxisError="未取得";

    private String mes="";

    private int count=0;
    private long[] lotList = new long[1000];
    private long[] latList = new long[1000];
    private int base=0;
    private int bai=1;
    private boolean flag=false;

    private Image      imgOff;
    private Graphics graOff;

    private      Command      exitCmd      =      new

```



```

Command("Exit",Command.SCREEN,0);
    private    Command    startCmd    =    new    Command("    開    始
",Command.SCREEN,1);

    private                                           Font
f=Font.getFont(Font.FACE_MONOSPACE,Font.STYLE_PLAIN,Font.SIZE_SMAL
L);

    private int fh;
    private int fx,fy;

    LocationCanvas(MsGpsLocationTest main){
        int i;
        this.main = main;

        fx=getWidth();
        fy=getHeight();

        imgOff = Image.createImage(fx,fy);
        graOff = imgOff.getGraphics();
        graOff.setFont(f);

        fh=f.getHeight();

        setCommandListener(this);
        addCommand(exitCmd);
        addCommand(startCmd);

        for(i=0;i<1000;i++){
            lotList[i]=0;
            latList[i]=0;
        }
    }

    public void locationUpdated(GpsLocation location){
        this.location = location;
        try{
            datum = location.getDatum();

```

```

        unit = location.getUnit();
        lon = location.getLon();
        lat = location.getLat();

        version = ""+location.getVersion();
        altitude = location.getAltitude();
        time = ""+(location.getTime()).getTime();
        fixMode = location.getFixMode();

        majorAxisAngle = location.getMajorAxisAngle();
        verticalError = location.getVerticalError();
        semiminorAxisError = location.getSemiminorAxisError();
        semimajorAxisError = location.getSemimajorAxisError();

        lotList[count]=toLong(lon);
        latList[count]=toLong(lat);
        count= (count+1)%1000;
        mes="";
    }catch(Exception e){
        if(location==null)mes="ぬるぽ";
        else mes=e.toString();
    }
    repaint();
}

public synchronized void paint(Graphics g){
    int i;
    graOff.setColor(0xFFFFFFFF);
    graOff.fillRect(0,0,fx,fy);

    graOff.setColor(0x00FF00);
    for(i=0;i<1000;i++){
        if(lotList[i]==0 || lotList[i+1]==0)break;
        graOff.drawLine((int)(fx/2+(lotList[i]-lotList[base])/bai),
            (int)(fy/2-(latList[i]-latList[base])/bai),
            (int)(fx/2+(lotList[i+1]-lotList[base])/bai),
            (int)(fy/2-(latList[i+1]-latList[base])/bai) );
    }
}

```

```

    }

    graOff.setColor(0);
    graOff.drawString("測位系 : " + datum ,0,0,g.LEFT|g.TOP);
    graOff.drawString("座標系 : " + unit ,0,fh,g.LEFT|g.TOP);
    graOff.drawString("経度 : " + lon ,0,fh*2,g.LEFT|g.TOP);
    graOff.drawString("緯度 : " + lat ,0,fh*3,g.LEFT|g.TOP);

    graOff.drawString(" バ ー ジ ョ ン : " + version ,
0 ,fh*4,g.LEFT|g.TOP);
    graOff.drawString("高度 : " + altitude , 0, fh*5 , g.LEFT|g.TOP);
    graOff.drawString("時間 : " + time ,0,fh*6 ,g.LEFT|g.TOP);
    graOff.drawString("測位方法 : " + fixMode ,0,fh*7 ,g.LEFT|g.TOP);

    graOff.drawString(" 長 軸 角 度 : " + majorAxisAngle ,
0 ,fh*8,g.LEFT|g.TOP);
    graOff.drawString(" 高度誤差 : " + verticalError , 0 , fh*9 ,
g.LEFT|g.TOP);
    graOff.drawString(" 短 軸 誤 差 : " + semiminorAxisError ,
0 ,fh*10 ,g.LEFT|g.TOP);
    graOff.drawString(" 長 軸 誤 差 : " + semimajorAxisError ,
0 ,fh*11 ,g.LEFT|g.TOP);

    if(mes.equals("")){
        graOff.drawString("倍率:" +bai +" 中心:" +base +" 数 :
"+count, 0 ,fy ,g.LEFT|g.BOTTOM);
    }else{
        graOff.drawString(mes, 0 ,fy ,g.LEFT|g.BOTTOM);
    }
    g.drawImage(imgOff,0,0,g.LEFT|g.TOP);
}

public void commandAction(Command c,Displayable s) {
    if(c==exitCmd){
        main.doExit();
    }else if(c==startCmd){
        GpsLocation.setLocationListener(this,0);
    }
}

```

```

    }
    repaint();
}

private long toLong(String sd){
    int dd,ms;
    boolean fs;
    fs=true;
    if(sd.startsWith("+")) sd = sd.substring(1);
    if(sd.startsWith("-")) {
        sd = sd.substring(1);
        fs=false;
    }
    try{
        int index = sd.indexOf(".");
        dd = Integer.parseInt(sd.substring(0,index));
        ms = Integer.parseInt(sd.substring(index+1));
    }catch(Exception e){
        mes=e.toString();
        return -1;
    }
    if(fs) return (dd*(1000000)+ms);
    else return (-1)*(dd*(1000000)+ms);
}

protected synchronized void keyPressed(int keyCode) {
    if(keyCode==0){
        mes="";
        repaint();
        return;
    }
    int action=getGameAction(keyCode);
    if(action==UP){
        if(bai<10000000)bai*=2;
    }else if(action==DOWN){
        if(bai>1)bai/=2;
    }else if(action==LEFT){
        base=(base+999)%1000;
    }
}

```

```
        }else if(action==RIGHT){
            base=(base+1)%1000;
        }
        repaint();
    }
}
```


GPS 内蔵型携帯電話で取得した位置を Web 上で表示させる方法

```
#!/usr/local/bin/perl
#
# imakoko.cgi --- au の GPS 携帯から位置情報を連続的に取得し、GoogleMap に
#               プロット、軌跡を描く cgi スクリプト
#
# 設置方法：
# 1. 設置するディレクトリでの GoogleMapsAPIKey を取得する。
# 2. この cgi の「各サイトでの個別設定」の部分を設定する。
# 3. この cgi を設置する。(cgi スクリプト自体の文字コードは utf-8 を推奨)
# 4. 設置するディレクトリ、このプログラムのパーミッションを適切に設定する。
#
### ここから、各サイトでの個別設定

# シェバング(最初の行)は、OK ですか?【確認必須】

# この cgi を設置するディレクトリでの GoogleMapsAPIKey【設定必須】
$GoogleMapsAPIKey = "";
#$GoogleMapsAPIKey
'ABQIAAAApBUCPi6dZcJJChQ2Jf06RRQKKSeFISyvZfdbrKeuaYSwEyElfRTQsr4PH
4yQ9sjKQb6Rd9BDhSBeYg';

# 日本語のコード (utf-8(非変更)を推奨。それ以外は動作未確認。)

$Charset = 'utf-8';
#$Charset = 'euc';

# 測地系 (WGS-84(非変更)を推奨。それ以外は動作未確認。)

( $Datum , $DatumCode ) = ( 'wgs84' , 0 );
#( $Datum , $DatumCode ) = ( 'tky' , 1 );

# 定数の設定 (日本語で表示したければコメントを付け替える)

( $DEG , $MIN , $SEC , $DirN , $DirS , $DirE , $DirW , $sALT , $sDAT , $sACC ) = ( '°' ,
```

```

'/' , '/' , 'N' , 'S' , 'E' , 'W' , 'Altitude' , 'Datum' , 'Accuracy');
#( $DEG , $MIN , $SEC , $DirN , $DirS , $DirE , $DirW , $sALT , $sDAT , $sACC ) = ( '
度' , '分' , '秒' , '北緯' , '南緯' , '東経' , '西経' , '高度' , '測地系' , '測位精度');

# 設置場所関連 (非変更を推奨)

$ConfigFile = './config.txt';
$LogDir = '.';

# URL は自動取得。サーバによってうまくいかない場合は、ここを変更。

$U="http://$ENV{HTTP_HOST}$ENV{SCRIPT_NAME}";

# device:gpsone を呼び出す URI

$R
=
"device:gpsone?url=$U&ver=1&datum=$DatumCode&unit=0&acry=0&number=0";

### メインルーチン

# 引数の取得と判断

$_ = $ENV{QUERY_STRING};

if ( /^(\log.*)/ ) {
    # ログファイル名であれば、地図を表示
    putGoogleMap( $1 );
    # エラーになって戻ってきたら、標準画面を表示
    putDefaultMenu();
} elseif ( /lat=/ && /¥&/ ) {
    # device:gpsone が吐き出すものであれば
    readConfig();
    if ( $1 ne " " ) { $1 = '.' . $1 }
    if ( -e "log$1" ) {
        open( LOG , "+>>$LogDir/log$1" );
        seek( LOG , 0 , 2 );
    }
}

```

```

    } else {
        open( LOG , ">$LogDir/log$l" );
    }
    print LOG "$_¥n";
    close LOG;
    putHdml();
} elsif ( /^menu/ ) {
    # ログを選ぶメニュー
    print "Content-Type: text/html¥n¥n";
    print "<html><head><title>imakoko.cgi --- LogMenu</title></head>";
    print "<body>imakoko.cgi --- LogMenu<br><hr>¥n";
    opendir( DIR , "$LogDir" );
    @list = sort readdir( DIR );
    closedir( DIR );
    foreach( @list ) {
        if ( /^log/ ) { print qq(<a href="$U?$_">$_</a><br>¥n); }
    }
    print "</body></html>¥n";
    exit;
} else {
    # 引数が無かったら、または不適なものであればメニューを表示
    putDefaultMenu( $_ );
}
exit;

```

サブルーチン

putDeafultMenu --- 標準メニューを表示

```

sub putDefaultMenu {
    my( $s ) = @_ ;
    $s .= '&';
    if ( $s =~ /name=([^\&]*)&/ ) {
        my( $k ) = $1;
        if( $k =~ /[^\_0-9a-zA-Z]/ ) {
            $wl = "$k is not valid name. Write in [_0-9a-zA-Z]";

```

```

        readConfig();
    } else {
        $l = $k;
    }
}
if ( $s =~ /intv=([0-9]*)&/ ) {
    $v = $1;
    writeConfig();
} else {
    readConfig();
}
putHtml();
}

```

readConfig ... ログ名、ログ間隔を設定ファイル (\$ConfigFile) から読み出す

```

sub readConfig {
    if ( -e $ConfigFile ) {
        open( IN , $ConfigFile );
        my( @r ) = <IN>;
        close IN;
        foreach( @r ) {
            if ( /LogName=([_0-9a-zA-Z]*)/ ) { $l = $1 }
            if ( /Interval=([0-9]+)/ ) { $v = $1 }
        }
    } else {
        $l = ""; $v = 600;
    }
}

```

writeConfig ... ログ名、ログ間隔を設定ファイル (\$ConfigFile) に保存する
 # エラー処理が甘い？

```

sub writeConfig {
    open( OUT , ">$ConfigFile" );
    print OUT "LogName=$l¥nInterval=$v¥n";
}

```

```

        close OUT;
    }

# putHtml --- メニューの HTML 部分を表示する
# $U : url identified me , $R : gpsone , $l : LogName , $wl : About LogName
# $v : interval ( * 0.1 sec. )

sub putHtml
{
    print "Content-type: text/html¥n¥n";
    print <<"END_OF_HTML";
    <html>
    <head>
    <title>imakoko.cgi</title>
    </head>
    <body>
    imakoko.cgi --- Main Menu<br>
    <hr>
    <a href="$R">StartLog</a><br>
    <form method="get" action="$U">
    LogConfig<br>
    LogName:<input type="text" name="name" size="10" value="$l">$wl<br>
    Interval:<input type="text" name="intv" size="6" value="$v">* 0.1 sec.<br>
    <input type="submit" value="Execute">
    </form>
    <a href="$U?menu">ReadLog</a><br>
    </body>
    </html>
    END_OF_HTML
    exit;
}

# putHdml --- au 携帯電話に渡す HDML 部分を表示する

sub putHdml {
    my( $s ) = @_;
```



```

my $date;
$date = localtime(time);
print "Content-Type: text/x-hdml¥n¥n";
print <<"END_OF_HDML";
<HDML VERSION=3.1 TTL=0>
<DISPLAY TIMERDEFAULT=$v>
<ACTION TYPE=ONTIMER TASK=GO DEST="$R">
imakoko.cgi<BR>$date<BR>
<A TASK=GO DEST="$R">Logging Now</A><BR>
<A TASK=GO DEST="$U">Back to MainMenu</A>
</DISPLAY>
</HDML>
END_OF_HDML
exit;
}

```

putGoogleMap --- ログを GoogleMap 上に表示する

```

sub putGoogleMap
{
# ログの取得
my( $s ) = @_ ;
if ( !e $s ) { return; }
open( FILE , $s );
@logBody = <FILE>;
close FILE;
# ログの処理
foreach( @logBody ) {
    s/%([A-Fa-f0-9][A-Fa-f0-9])/pack("c", hex($1))/ge;
    # 引数を変数名として取得
    foreach $key ( lat , lon , alt , 'time' , datum , vert , fm ) {
        if ( /$key=( [^&]* ) / ) { $$key = $1 } else { $$key = "" }
    }
    if ( ( $lat eq "" ) || ( $lon eq "" ) ) {
        # lat , lon が無ければ無視
        next;
    }
}

```

```

    } else {
        foreach $key ( lat , lon ) {
            $_ = $$key;
            s/([0-9]+)°([0-9]+)'([0-9]+)"/$1$DEG$2$MIN$3$SEC/;
            $$key = $1 + $2 / 60 + $3 / 3600;
            $skey = 's' . $key;
            $$skey = $_;
        }

        $slat =~ s/^°+/$DirN/;
        $slat =~ s/^°+/$DirS/;
        $slon =~ s/^°+/$DirE/;
        $slon =~ s/^°+/$DirW/;

        $time =~ s/(\d\d\d\d)(\d\d)(\d\d)(\d\d)(\d\d)(\d*)| $1/$2/$3
$4:$5:$6|;

        $fm = ( 'GPS-FIX' , 'Hybrid-FIX' , 'AFLT-FIX' ,
'SECTOR-CENTER' )[ $fm ];
        $datum = ( 'WGS-84' , 'TKY' )[ $datum ];

        $s = " $slat¥n $slon¥n $sALT:$salt
m¥n $time¥n $sDAT:$datum¥n $sACC:$fm¥n";

        $s =~ s/¥n/<br>/g;
        $s = qq(<div style="width:200px;font-size:9pt;">$s</div>);
        push( @L1 , qq| new GPoint( $lon , $lat ), '$s' | );
        push( @L2 , qq| new GPoint($lon,$lat)| );
        $clat = $lat; $clon = $lon;
    }
}

# GoogleMap の描画

print "Content-type: text/html; charset=$Charset¥n¥n";

## ヘッダ部分と当初部処理
print <<"END_OF_HEAD";
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html                                xmlns="http://www.w3.org/1999/xhtml"
xmlns:v="urn:schemas-microsoft-com:vml">
<head>
<meta http-equiv="content-type" content="text/html; charset=$Charset"/>
<style type="text/css">
v\:*{
    behavior:url(#default#VML);
}
</style>
<script
src="http://maps.google.com/maps?file=api&v=1&key=${GoogleMapsAPIKey}&datum=
$Datum" type="text/javascript"></script>
</head>
<body>
    <div id="map" style="width: 600px; height: 600px"></div>
    <script type="text/javascript">
        //

        var map = new GMap(document.getElementById("map"));
        map.addControl(new GLargeMapControl());
        map.addControl(new GMapTypeControl());
        map.centerAndZoom(new GPoint( $clon , $clat ) , 1 );
</pre>
</div>
<div data-bbox="137 621 282 638" data-label="Text">
<p>END_OF_HEAD</p>
</div>
<div data-bbox="135 664 384 680" data-label="Text">
<p>## ログ地点のマーカーを作成</p>
</div>
<div data-bbox="135 686 628 743" data-label="Text">
<pre>
foreach( @L1 ) {
    print "  map.addOverlay( createMarker( $_ ));\n";
}
</pre>
</div>
<div data-bbox="135 770 484 787" data-label="Text">
<p>## ログ地点を結ぶ線分(ポリライン)の作成</p>
</div>
<div data-bbox="135 812 374 830" data-label="Text">
<p># 線分の座標を表す配列作成</p>
</div>
<div data-bbox="135 835 368 873" data-label="Text">
<pre>
print "  var points = [];\n";
foreach( @L2 ) {
</pre>
</div>
```

```

        print "  points.push( $_);¥n";
    }
# ポリラインの描画
print "  map.addOverlay(new GPolyline(points,'#000000',5,0.7));¥n";

## マーカーの作成で使った createMarker の定義と末尾処理
print <<"END_OF_FOOT";

    function createMarker( point , html ) {
        var marker = new GMarker( point );
        GEvent.addListener(marker, 'click', function() {
            marker.openInfoWindowHtml(html);
        });
        return marker;
    }

//]]>
</script>

```